Identifying and Analyzing Long-term Dynamical Behaviors of Gene Regulatory Networks with Hybrid Modeling

Honglu Sun

MÉMOIRE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES

ÉCOLE DOCTORALE N° 641
Mathématiques et Sciences et Technologies du numérique, de l'Information et de la Communication
Spécialité : *Informatique*

Par

Honglu SUN

Identifying and Analyzing Long-term Dynamical Behaviors of Gene Regulatory Networks with Hybrid Modeling

Projet de recherche doctoral présenté et soutenu à l'Ecole Centrale de Nantes, le 18 décembre 2023
Unité de recherche : UMR 6004, Laboratoire des Sciences du Numérique de Nantes (LS2N)

Rapporteurs avant soutenance :

Thao DANG	Directrice de recherche CNRS, Université Grenoble Alpes
Ovidiu RADULESCU	Professeur des universités, Université de Montpellier

Composition du Jury :

Président :	Jean-Luc GOUZE	Directeur de recherche, INRIA Côte d'Azur
Examinateurs :	Thao DANG	Directrice de recherche CNRS, Université Grenoble Alpes
	Ovidiu RADULESCU	Professeur des universités, Université de Montpellier
	Hélène COLLAVIZZA	Maître de conférences HDR, Université Côte d'Azur
Directeur de recherches doctorales :	Morgan MAGNIN	Professeur des universités, École Centrale de Nantes
Co-enc. de recherches doctorales :	Maxime FOLSCHETTE	Maître de conférences, École Centrale de Lille

Acknowledgement

First and foremost, I would like to express my since regratitude to my supervisors Prof. Morgan MAGNIN and Assoc. Prof. Maxime FOLSCHETTE for giving me this very interesting research topic and for the continuous support to my PhD study. This kind support covered many aspects and also came to my personal life, which solved many problems in the laboratory or in my life, and allowed me to be able to spend more time on my research works. I also like to appreciate that they allowed me to explore on my interests which leads to a very interesting and exciting PhD experience.

Besides my advisors, I would also like to thanks all my jury members who have kindly devoted their time to my thesis. I would like to thank Dr. Thao DANG and Prof. Ovidiu RADULESCU for accepting to be my rapportrice / rapporteur, and thank Dr. Jean-Luc GOUZE and Assoc. Prof. Hélène COLLAVIZZA for accepting to be my examinateur / examinatrice.

My sincere thanks also goes to my other kind team members: Prof. Olivier ROUX, Dr. Tony RIBEIRO, Dr. Samuel BUCHET and Mr. Mitsuhiro ODAKA. Without their help, it would not be possible for me to conduct this research.

I am also deeply indebted to Prof. Jean-Paul COMET and Dr. Nathalie THERET for accepting to be my CSI members. Their recognition of my works gave me a lot of motivations.

During my thesis, also thanks to my advisors, I had the opportunities to exchange with many colleagues from other teams or other laboratories: Prof. Jean-Paul COMET, Prof. Gilles BERNOT, Assoc. Prof. Denis PALLEZ, Assoc. Prof. Benoit DELAHAYE, Assoc. Prof. Guillaume CANTIN and Dr. Laetitia GIBART. They gave me many great inspirations and suggestions on my works.

I gratefully acknowledge the funding provided by China Scholarship Council (CSC) that made my PhD work possible.

Last but not the least, I would like to thank my parents and my girl friend Yizheng for their love and encouragement.

Table of Contents

1 Introduction **9**

1.1 Context and Motivations 9

1.2 Contributions 12

1.3 Collaborations 15

1.4 Organization of the Manuscript 15

1.5 Notations 15

2 State of the Art of Gene Regulatory Networks Modeling **17**

2.1 Ordinary Differential Equations (ODE) 17

2.2 Discrete Modeling 18

2.3 Hybrid Modeling 20

2.3.1 Piecewise Affine Systems (PWA) 21

2.3.2 Hybrid Gene Regulatory Networks (HGRN) 21

3 Search and Stability Analysis of Limit Cycles **27**

3.1 Limit Cycles of HGRNs 27

3.2 Search of Periodic Hybrid Trajectories 30

3.2.1 Abstraction of HGRN with Discrete Domains 31

3.2.2 Search of Closed Discrete Trajectories 33

3.2.3 Identification of Periodic Hybrid Trajectories with Poincaré Map . . 39

3.3 Stability Analysis of Limit Cycles 42

3.3.1 Continuity in the Neighborhood of Periodic Hybrid Trajectories . . 42

3.3.2 Eigenanalysis 55

3.4 Applications 57

3.4.1 3-dimensional Models of Negative Feedback Loop 58

3.4.2 5-dimensional Model of Cell Cycle 61

3.5 Summary 64

4 Reachability Analysis **67**
4.1 Problem Statement . . . 67
4.2 Different Classes of Hybrid Trajectories . . . 70
4.2.1 Hybrid Trajectories Halting in Finite Time . . . 70
4.2.2 Hybrid Trajectories Attracted by Cycles of Discrete Domains . . . 70
4.2.3 Chaotic Hybrid Trajectories . . . 78
4.3 Reachability Analysis Algorithm . . . 81
4.4 Application: Estimation of Basins of Attraction . . . 87
4.5 Summary . . . 88

5 Condition for Sustained Oscillation in Canonical Repressilator **91**
5.1 Introduction of Canonical Repressilator . . . 91
5.2 A HGRN of Canonical Repressilator . . . 92
5.3 Qualitative Behaviors in this HGRN of Canonical Repressialtor . . . 94
5.4 A Sufficient and Necessary Condition for Sustained Oscillation . . . 98
5.4.1 Symbolic Computation of Poincaré Map . . . 98
5.4.2 Symbolic Eigenanalysis . . . 101
5.5 Computation of Sufficient Separable Constraints on Parameters . . . 104
5.5.1 Condition Simplification . . . 105
5.5.2 Satisfiability under Separable Constraints . . . 106
5.5.3 Search of Separable Constraints . . . 108
5.6 Summary . . . 110

6 Parameter Identification based on Time Series Data **111**
6.1 Definition of the Parameter Identification Problem . . . 111
6.1.1 Representation of the Parameter Set to Identify . . . 111
6.1.2 Data Representation . . . 113
6.2 Description of the Method . . . 115
6.2.1 Objective Function for the Optimization of Regular Parameter Set . 115
6.2.2 Estimation of Potential Parameter Set . . . 115
6.2.3 Parameter Identification based on Genetic Algorithm . . . 117
6.3 Application . . . 120
6.4 Summary . . . 122

7 Condition for a Discrete Periodic Attractor in 4-dimensional Repressilators **123**

7.1 4-dimensional Repressilators 123

7.2 Feature Selection and Search of Candidate Condition based on Decision Tree 128

7.3 Condition Simplification 131

7.4 Number of Oscillatory Dimensions in a Discrete Periodic Attractor 134

7.5 Summary 134

8 Conclusion and Perspectives **137**

8.1 Summary of Contributions and Limits 137

8.2 Perspectives 139

Bibliography **141**

Chapter 1

Introduction

1.1 Context and Motivations

Gene Regulatory Networks

Gene expression has two major steps: Transcription, which generates mRNA from genes, and translation, which generates proteins from mRNA. In general, each mRNA molecule produces a specific protein (or set of proteins). This protein can be structural (giving it a particular structural property) or be an enzyme (catalyzing a certain reaction). Additionally, it can also serve only to activate or inhibit other genes. These regulations (activation or inhibition) between genes constitute *gene regulatory networks*, which play a key role in various cellular processes and pathways.

One major problematic of gene regulatory networks is analyzing their dynamical properties. In the literature, different kinds of dynamical properties have been used to describe the behaviors of gene regulatory networks, such as the existence of stable states or oscillations, the reachability between state(s) (for example, whether the trajectory/trajectories from a state can reach certain state/states), or some properties described by temporal logic (for example, the sequence of states on a trajectory always satisfies certain property, or will eventually satisfy certain property, or will satisfy certain property until another property is satisfied). Knowing the possible dynamical properties of gene regulatory networks can help us understand better the underlying nature of these biological systems, and can further guide us to develop control methods to change the behaviors of certain systems for medical purposes. Actually, the complexity of gene regulatory networks can increase dramatically when the number of genes increases, which makes it hard to analyze these properties.

Modeling of Gene Regulatory Networks

In order to analyze these dynamical properties, one solution is to model mathematically the gene regulatory networks. Such approaches can be more efficient and cheaper compared to biological experiments, and can also allow us to understand the dynamics of the networks at system level. First, a gene regulatory network is abstracted by a directed graph called *influence graph*, where the vertices represent the genes in the system and the arcs represent the regulations between genes. In fact, the initial understanding of gene regulatory networks normally only contains the relations between genes, so it is natural that they are abstracted by directed graphs as a first step. As stated before, a gene can generate different products (mRNA and protein), while in an influence graph we usually abstract all products of one gene by one entity (one vertex). The structure of the influence graphs can give some insights about the dynamical properties of gene regulatory networks, but to determine the dynamical properties, influence graphs are not enough, dynamical models are also required.

In the literature, different modeling frameworks have been applied to model gene regulatory networks, mainly continuous models [1–3], discrete models [4–9] and hybrid models [10–12]. In continuous models, a state is a real vector representing the continuous expression of each gene. In discrete models, the continuous state space is divided into discrete regions and a state, also called a discrete state, is a vector of integers representing a discrete region. Hybrid models have both continuous and discrete components. In hybrid models, like in discrete models, the continuous state space is divided into discrete regions, but contrary to discrete models where we do not consider the dynamics inside these discrete regions, the dynamics of hybrid models inside each discrete region is described by a simplified continuous model. More details about these modeling frameworks are introduced in Chapter 2.

One major difference between discrete models and models having continuous behaviors (continuous or hybrid models) is that: Discrete models can be applied on systems of large scale, but, for continuous or hybrid models, it is generally difficult to study dynamical properties of models in high dimension, mainly due to their dynamical complexities.

In fact, the dynamics of a discrete model can be described by all the possible transitions between discrete states, which is a directed graph with a finite number of nodes, so theoretically we can determine all dynamical properties, the only constraint is that, when the dimension (the number of genes) increases, the computational cost can be too high. Moreover, there are less parameters in discrete models (there are only parameters that

describe the qualitative properties of the system), so it requires less data or biological knowledge to construct the models.

For continuous models like ordinary differential equations, there are many open questions related to dynamical analysis, for instance, there is no general method to prove the existence of a stable limit cycle for models that have more than three dimensions. Some hybrid models can be considered as simplifications of ordinary differential equations, and their dynamical properties are easier to analyze, but still, the dimension of the systems that can be analyzed by these hybrid models is limited. Additionally, parameter identification of continuous or hybrid models requires more data.

Scope of the Study and Motivations

A recent proposed hybrid modeling framework: *Hybrid gene regulatory network (HGRN)* [11], which is an extension of Thomas' discrete modeling framework [5, 6], has lower dynamical complexity compared to other hybrid models. In HGRNs, the time derivative is a constant vector in each discrete region. So it is easier to analyze and could be potentially applied to reveal continuous dynamical properties of larger (compared to networks that are modeled by continuous models or other hybrid models) gene regulatory networks. Moreover, HGRN has a special dynamical behavior: The existence of sliding modes, which means that if a trajectory reaches a boundary of a discrete region but can not cross this boundary, then it slides along this boundary (such boundaries are also called "black walls" in the literature). The physical meaning of sliding modes is that some genes reach temporal stable stables, so their values will not change for some time, while the values of other genes keep changing.

Before this thesis, few works had studied analysis methods on HGRNs. Only in [11], a necessary and sufficient condition for the existence of a limit cycle of a specific 2-dimensional HGRN has been discussed. There is no general analysis method of N-dimensional HGRNs. So the study of such analysis methods is of great interest, which is the main research direction of this thesis.

Among different kinds of dynamical properties to describe gene regulatory networks, we mostly focus in this thesis on long-term dynamical properties, in other words, the behaviors of trajectories when time approaches infinity. For instance, a trajectory can converge to a limit cycle when time approaches infinity. Such limit cycles represent stable oscillations which can be found in many gene regulatory networks, such as the circadian clock. In discrete models, long-term dynamical properties correspond to discrete attrac-

tors. One limit is that, based on discrete models, it is hard to derive continuous behaviors inside discrete attractors; for example, we can not distinguish sustained oscillations and damped oscillations, while such continuous behaviors can be derived from HGRNs. The study of such long-term behaviors of gene regulatory networks is of great concern. Indeed, the stable limit cycles correspond to robust oscillations (oscillations will always recover after small perturbations) which exist in some genetic systems, for example circadian clock, and the analysis of limit cycles can help us verify whether the networks are correctly constructed.

1.2 Contributions

The major contributions of this thesis lie in both general analysis methods and the analysis of some specific networks. The contributions for the general analysis methods include a limit cycle analysis method presented at CMSB 2022 as regular paper [13] and a reachability analysis method presented at RP 2023 as regular paper [14].

General Analysis Methods

Limit Cycle Analysis

Our limit cycle analysis method of HGRNs can automatically identify limit cycles of HGRNs in any dimension and analyze their stability. This method can find all limit cycles that do not cross the same discrete region for more than once in one period. It is indeed important to analyze limit cycles, because many gene regulatory networks have stable oscillations which correspond to stable limit cycles. In this work, we also propose new concepts associated to HGRNs (for instance, in order to facilitate the analysis of HGRNs, we propose the new concept "discrete domain", based on which the dynamics of the original system can be abstracted by transitions on a directed graph), which can be useful for the future studies of HGRNs.

In the literature, even though few works exist about limit cycles analysis in HGRNs (only a 2-dimensional limit cycle was analyzed in [11]), limit cycles were studied in other hybrid models of gene regulatory networks. Most of these works are based on the Poincaré map, which is also the main idea of our approach. In [15, 16], the Poincaré map is used to study the limit cycle of simple piecewise affine systems in two dimensions. In these works, since the system is planar, it is easy to compute and analyze the Poincaré map. In [17–19],

methods are proposed to find and analyze limit cycles in higher dimensions of piecewise affine system with a uniform decay rate. The hypothesis of a uniform decay rate in these works makes it always possible to calculate a Poincaré map because they have a simple shape. However, for a general piecewise affine system, it is difficult to prove theoretically the existence of limit cycle except for some particular examples such as negative loops [20, 21].

Compared to previous works about limit cycles in hybrid systems, our work has two major novelties: (1) We consider limit cycles with sliding modes, and (2) We use an abstraction method in order to find automatically cycles of discrete regions, which might contain limit cycles (in other words, cycles of discrete regions which contain at least one continuous trajectory).

Reachability Analysis

The second proposed general analysis method is a reachability analysis algorithm of HGRNs, which can verify automatically if the trajectory from a singular state can eventually reach certain region (a set of states). This algorithm always stops in finite time, and in most cases, it gives the correct answer of this reachability problem. In fact, only in some particular cases, for example in the presence of chaos, the algorithm remains inconclusive. Such cases exist for general HGRNs, but they are not identified in any pre-existing HGRN of gene regulatory network. In fact, we have only identified one hybrid system which is equivalent to a HGRN and which has chaotic attractor, and this is a hybrid system of an electric circuit.

Actually, this work is related to the decidability problem, which is whether we can find an algorithm that always stops in finite time and answers this reachability problem. The decidability problem has not been investigated for HGRNs, but it has been studied among hybrid systems that are close to HGRNs. The hybrid systems that are the closest to HGRNs are piecewise-constant derivative systems (PCD systems), which are decidable in 2 dimensions [22] and become undecidable in 3 dimensions [23]. There exists also a class of decidable hybrid systems called initialized rectangular automata which does not include HGRNs. So whether HGRNs are decidable or not is still an open question. In our work, we formalize a set of trajectories for which this reachability problem is decidable. Meanwhile, for other trajectories, it is unknown.

Analysis of Specific Networks

Besides general analysis methods, we also analyze some specific networks of interest, called the repressilators, which led to two publications: The first one is about the 3-dimensional repressilator and has been presented at BIOINFORMATICS 2023 as regular paper [24]; the second one is about 4-dimensional repressilators and has been presented at CMSB 2023 as regular paper [25].

3-dimensional Repressilator

The 3-dimensional canonical repressilator is a widely studied network in the field of synthetic biology. In this work, we find a sufficient and necessary condition for the existence of sustained oscillations in a HGRN of 3-dimensional canonical repressilator, and we propose a method to compute simple constraints on parameters that satisfy this condition. This result could potentially contribute to the biological construction of synthetic repressilators. In the literature, most studies of the canonical repressilator use ordinary differential equations. This work is the first one based on HGRNs.

4-dimensional Repressilators

In our study of 4-dimensional repressilators in the form of discrete models, we provide a sufficient and necessary condition, which is described by the topological features of the influence graphs, for the existence of a discrete periodic attractor in 4-dimensional repressilators. This condition corresponds to a sufficient condition for the existence of oscillations in associated HGRNs. Other theoretical works investigate the relation between the topology of influence graphs and the dynamical properties. The major novelty of this work is that we discover a new topological feature.

Parameter Identification

Finally we also investigate parameter identification problems, which is the first step toward the understanding of systems. Indeed our analysis methods can be applied only after that the parameters are identified. There are different kinds of parameter identification problems that are related to gene regulatory networks, for example, there are works aiming to identify the regulations between genes, and there are other works that want to identify the parameters of dynamical models assuming that the influence graph is known. Our work belongs to the second case and is based on time series data.

1.3 Collaborations

The analysis work of the 3-dimensional repressilator and a part of the analysis work of the 4-dimensional repressilators come from a collaboration with the team of Jean-Paul Comet at the I3S laboratory (Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis) in Nice. For this collaboration, I stayed 3 weeks at I3S in June 2022.

1.4 Organization of the Manuscript

Chapter 2 introduces different modeling frameworks that have been used to model gene regulatory networks, including HGRNs. The notations used in this chapter are also used in the rest of this thesis.

Chapters 3 and 4 introduce our general analysis method of HGRNs: In Chapter 3 we present our limit cycle analysis method and in Chapter 4 we present our reachability analysis method. Some new concepts of HGRNs are introduced in Chapter 3, and are also used in the work of Chapter 4.

In Chapters 5, 6 and 7, we focus on some specific networks. Chapter 5 introduces our analysis of a HGRN of 3-dimensional repressilator, where we find conditions for the existence of sustained oscillations in this HGRN. This work relies on some methods of Chapter 3. In Chapter 6, we present a parameter identification method of HGRNs using the HGRN of repressilator in Chapter 5 as a case study. The work of Chapter 7 is a continuation of the work of Chapter 5: In Chapter 7, we present conditions for the existence of a discrete periodic attractor in discrete models of 4-dimensional repressilators.

Finally, Chapter 8 concludes the whole thesis.

1.5 Notations

d_s or d_{s_i}: A discrete state, introduced in Section 2.2

c_s or c_{s_i}: The celerity vector of a discrete state, introduced in Section 2.3.2

h or h_i: A hybrid state, introduced in Section 2.3.2

e or e_i: A boundary of HGRN, introduced in Section 2.3.2

τ or τ_i: A hybrid trajectory of HGRN, introduced in Section 2.3.2

$\mathcal{D}$ or $\mathcal{D}_i$: A discrete domain, introduced in Section 3.2.1

$\mathcal{T} = (\mathcal{D}_i, \mathcal{D}_{i+1}, ..., \mathcal{D}_{j-1}, \mathcal{D}_j)$: A sequence of discrete domains, introduced in Section 3.2.2

$M_{\mathcal{T}}$: The transition matrix of the sequence of discrete domains $\mathcal{T}$, introduced in Section 3.2.2

$\mathcal{S}$: A compatible zone, introduced in Section 3.2.2

$N_d(h, r)$: The neighborhood in the same discrete state of the hybrid state h with the radius r, introduced in Section 3.3

$N_{\mathcal{D}}(h, r)$: The neighborhood in the same discrete domain of the hybrid state h with the radius r, introduced in Section 3.3.1

$r(h)$: The reduction vector of hybrid state h, introduced in Section 3.3.2

$\mathcal{S}_r$: A reduction compatible zone, introduced in Section 3.3.2

Chapter 2

State of the Art of Gene Regulatory Networks Modeling

A gene regulatory network is abstracted mathematically by a directed graph $IG = (V, A)$ called *influence graph*, where V is the set of vertices representing the genes in the system and A is the set of arcs representing the regulations between genes. To understand the dynamical properties of gene regulatory networks, influence graphs are not enough, we also need dynamical models. Different classes of dynamical models have been used in the literature to model gene regulatory networks. The most used ones are ordinary differential equations (ODE) and discrete models, which are presented briefly in the first two sections of this chapter. Another modeling approach is hybrid modeling which is not yet widely used for now but is attracting more attention in recent years. In the third section of this chapter, the piecewise affine systems are presented at first, which are the most used hybrid systems to model gene regulatory networks; then the hybrid systems studied in this thesis, *Hybrid gene regulatory networks*, are presented.

2.1 Ordinary Differential Equations (ODE)

Ordinary differential equations (ODE) are mostly used in the literature to model gene regulatory networks [1–3, 26–28]. The use of ODEs is based on the assumption that the numbers of molecules describing the expressions of genes are very big so that real numbers can be used to represent the expressions of genes. A widely used form to describe the dynamics of one gene by ODEs, for example G_0, is given in Eq (2.1).

$$\frac{dx_0}{dt} = k_0 + f(x_1, x_2, ..., x_m) - \gamma x_0 \tag{2.1}$$

where x_i is a real number describing the expression value of gene G_i; $G_1, G_2, ..., G_m$ are the genes that influence gene G_0; k_0 is the production rate of gene G_0 when it is not regulated

by other genes; γ is the degradation rate which describes, for example, the efficacy of protease if the expression of the gene is measured by the amount of generated protein; function f describes the regulations of $G_1, G_2, ..., G_m$ on G_0, and is usually based on Hill function, for example f can be of the following form.

$$f(x_1, x_2, ..., x_m) = f_1(x_1) + f_2(x_2) + ... + f_m(x_m) \tag{2.2}$$

where $f_i(x_i) = k_i \frac{x_i^n}{\theta_i^n + x_i^n}$ if G_i activates G_0 and $f_i(x_i) = k_i \frac{\theta_i^n}{\theta_i^n + x_i^n}$ if G_i inhibits G_0.

There are two major limits of ODEs: 1) Their dynamics are hard to analyze; 2) There are many parameters to be identified. For now it is still difficult (or impossible) to analyze large networks based on ODEs.

2.2 Discrete Modeling

Discrete models are also widely used to model gene regulatory networks, which in fact attract more and more attention in the last 20 years. In discrete models [4–9], the continuous expression of a gene is abstracted by an integer (e.g. 0, 1, 2, ...), called discrete level, which describes the discrete expression level of a gene. We note a_i as the discrete level of gene G_i. A discrete model is a logic program which is a set of logic rules. The form of a logic rule is shown as follows:

$$a_i = k \leftarrow \phi_i \tag{2.3}$$

where k is a possible discrete level of G_i, and ϕ_i is a logic formula. The form of a logic formula ϕ is given as follows:

$$\phi \ :== \ \top \mid \bot \mid a_i \sim k \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \tag{2.4}$$

where k is a possible discrete level of gene G_i, $\sim$ is one of the relations $\{>, <, =, \geq, \leq\}$, ϕ_1 and ϕ_2 are also logic formulas.

A logic rule (see Eq (2.3)) indicates that if at discrete time t (t is an integer) ϕ_i is satisfied, then at time $t + 1$ the discrete level of G_i can be updated to k. For example, a

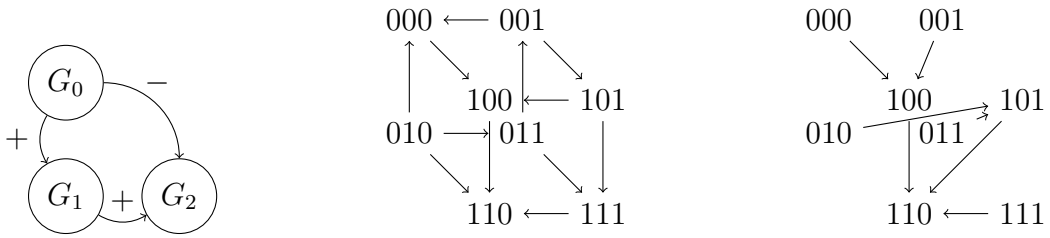


Figure 2.1 – Left: A 3-dimensional influence graph. Middle (resp. Right): A transition graph of discrete states with asynchronous (resp. synchronous) semantics, based on the influence graph on the left and the logic program of Eq. 2.5.

possible discrete model of the influence graph in Fig 2.1-Left is given as follows:

$$\begin{aligned} a_0 &= 1 \leftarrow \top \\ a_1 &= 1 \leftarrow (a_0 = 1) \\ a_1 &= 0 \leftarrow (a_0 = 0) \\ a_2 &= 1 \leftarrow (a_0 = 0) \wedge (a_1 = 1) \\ a_2 &= 0 \leftarrow (a_0 = 1) \vee (a_1 = 0) \end{aligned} \tag{2.5}$$

In this example, the second line indicates that, if at time t the discrete level of G_0 is 1, then at time $t+1$ the discrete level of G_1 can be updated to 1; this logic rule corresponds to the activation from G_0 to G_1 in the influence graph. The first line indicates that for any moment t, the discrete level of G_0 can be updated to 1 at time $t+1$ (in fact, once the discrete level of G_0 reaches 1, it will remain at 1).

The simulation of a discrete model is not solely dependent on these logic rules, but also relies on the semantics of the model. Intuitively, the semantics dictates the number of genes that can be updated simultaneously. Various semantics, such as synchronous, asynchronous, general and most permissive, have been proposed in the literature [29, 30]. For example, the synchronous semantics means that all genes are updated simultaneously and the asynchronous semantics means that only one gene can be updated at a time.

Based on the choice of semantics, we can obtain the transition graph of discrete states of a discrete model. A discrete state is an integer vector of length N (if there are N genes in the system), noted by d_s, which assigns the discrete level d_s^i to gene G_i, where $i \in \{1, 2, 3, ..., N\}$ and d_s^i is the i^{th} component of d_s. The transition graph of discrete states is a directed graph containing all possible transitions between discrete states from

t to $t+1$. The transition graph of discrete states describes the dynamics of a discrete model. Note that depending on the semantics, the dynamics can be non-deterministic, as is the case for the asynchronous semantics.

For example, Fig 2.1-Middle presents the transition graph of discrete states, derived by asynchronous semantics, of the discrete model described by the logic program in Eq (2.5). Consider the discrete state 101 (representing the assignment $a_0 = 1, a_1 = 0, a_2 = 1$), according to the logic program in Eq (2.5), G_1 can be updated from 0 to 1 and G_2 can be updated from 1 to 0. Since we use asynchronous semantics, only one gene can be updated, so it can reach 100 or 111, but it cannot make two updates at the same time to reach 110. Fig 2.1-Right presents the transition graph of discrete states, derived by synchronous semantics, of the same discrete model. It can be seen that the dynamics changes when the semantics changes.

Compared to ODEs, discrete models are easier to analyze and implement, while they can still capture the qualitative dynamical properties of gene regulatory networks, which is sometimes enough in certain biological studies. Due to the dynamical simplicity of discrete models, they can be used to study large networks. A part of this thesis focuses on discrete models (Chapter 7).

2.3 Hybrid Modeling

Hybrid models are dynamical systems that contain at the same time discrete and continuous components, which, following ODEs and discrete models, have been used to model gene regulatory networks [15–19]. Like discrete models, in hybrid models, the continuous state space is divided into several discrete regions, and the dynamics in each discrete region is described by simplified ODE. Compared to ODE, generally the dynamical properties of hybrid models are easier to analyze, and compared to discrete models, we can know the continuous behaviors of systems using hybrid models.

In the following subsections, firstly the *piecewise affine systems* are presented which are the most studied hybrid models of gene regulatory networks, then *hybrid gene regulatory networks* are presented which is the major formalism studied in this thesis.

2.3.1 Piecewise Affine Systems (PWA)

The piecewise affine systems (PWA) [15–19] are simplifications of ODEs. One major reason behind these simplifications is that the widely used Hill functions in ODEs can be approximated by step functions. A general form of PWAs to model gene regulatory networks is given in Eq (2.6).

$$\frac{dx_0}{dt} = k_0 + g(x_1, x_2, ..., x_m) - \gamma x_0 \tag{2.6}$$

which is similar to Eq (2.1), but here the function g is based on step functions, for example it can be of the following form:

$$g(x_1, x_2, ..., x_m) = g_1(x_1) + g_2(x_2) + ... + g_m(x_m) \tag{2.7}$$

where

$$g_i(x_i) = \begin{cases} k_i^+ & \text{if } x_i \geq \theta_i \\ k_i^- & \text{if } x_i < \theta_i \end{cases} \tag{2.8}$$

with k_i^+ and k_i^- real numbers.

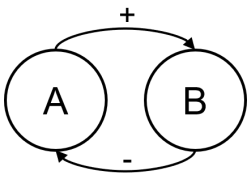
In fact, these θ_i divide the continuous state space into several discrete regions and in each discrete region the dynamics is described by affine functions.

2.3.2 Hybrid Gene Regulatory Networks (HGRN)

In this thesis, we mainly focus on a class of hybrid models called Hybrid gene regulatory networks (HGRN). The difference between HGRNs and PWAs mainly lies in the behaviors of trajectories on boundaries. HGRN and its basic notions are defined formally as follows.

Consider a gene regulatory network with N genes. For $i \in \{1, 2, 3, ..., N\}$, the i^{th} gene, noted G_i, has $n_i + 1$ discrete levels which are represented by integers: $\{0, 1, 2, ..., n_i\}$. The set of all discrete states is $E_d = \left\{d_s = (d_s^i)_{i\in[1;N]} \in \mathbb{N}^N \mid \forall i \in \{1, 2, ..., N\}, d_s^i \in \{0, 1, ..., n_i\}\right\}$. In fact, the notion of discrete state in HGRNs is similar as the one in discrete models.

Definition 1 (Hybrid gene regulatory network (HGRN)) *A* Hybrid gene regulatory network *(HGRN) is noted $\mathcal{H} = (E_d, c)$. E_d is the set of all discrete states. c is a function from E_d to $\mathbb{R}^N$. For each $d_s \in E_d$, $c(s)$, also noted c_s, is called the* celerity *of discrete state d_s and describes the time derivative of the system in d_s.*



A	B	C_A	C_B
0	0	0.6	−0.7
0	1	−0.7	−0.9
1	0	0.7	0.8
1	1	−0.6	0.9

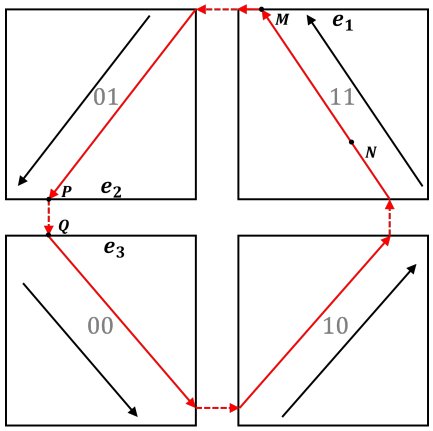


Figure 2.2 – Example of a HGRN in 2 dimensions. Left: Influence graph (negative feedback loop with 2 genes, $A = G_1$, $B = G_2$). Middle: Example of corresponding parameters (celerities). Right: Corresponding example of dynamics; abscissa represents gene A and ordinate represents gene B.

Definition 2 (Hybrid state of HGRN) *A* hybrid state *of a HGRN is a couple* $h = (\pi, d_s)$ *containing a* fractional part π*, which is a real vector in* $[0,1]^N$*, and a discrete state* d_s *in* E_d*.* E_h *is the set of all hybrid states.*

Definition 3 (Hybrid trajectory) *A hybrid trajectory* τ *is a function from a time interval* $[0, t_0]$ *to* $E_\tau = E_h \cup E_{sh}$*, where* $t_0 \in \mathbb{R}^+ \cup \{\infty\}$*,* E_h *is the set of all hybrid states, and* E_{sh} *is the set of all finite or infinite sequences of hybrid states:* $E_{sh} = \{(h_0, h_1, ..., h_m) \in (E_h)^{m+1} \mid m \in \mathbb{N} \cup \{\infty\}\}$.

A hybrid trajectory τ is called a *periodic hybrid trajectory* if it is defined on $[0, \infty[$ and $\exists T > 0, \forall t \in [0, \infty[, \tau(t) = \tau(t + T)$.

A 2-dimensional HGRN is shown in Fig 2.2. In this example, there are two genes, that we rename for simplicity: $A = G_1$ and $B = G_2$. Gene A activates gene B, so we can consider that there exists a threshold of gene A such that, when the expression value of gene A is above this threshold, the derivative of gene B is positive, and when it is below this threshold, the derivative of gene B is negative. Similarly for gene B, since B inhibits gene A, when the expression value of B is above a threshold, the derivative of gene A is negative, and when it is below this threshold, the derivative of gene A is positive. Since each gene has one threshold, they have two discrete levels: 0 and 1. Therefore, this gene regulatory network has four discrete states: 00, 01, 10 and 11, where, for example, 01 is a shorthand representing that the discrete level of gene A is 0 and the discrete level of gene B is 1.

In the illustration of the state space (Fig 2.2-Right), black arrows represent the celerities (time derivatives) of each discrete state and red arrows represent a possible hybrid trajectory of this system, which happens, in this particular case, to be a periodic hybrid trajectory.

Definition 4 (Boundary) *A boundary in a discrete state d_s is a set of hybrid states defined by $e(G_i, \pi_0, d_s) = \{(\pi, d_s) \in E_h \mid \pi^i = \pi_0, \}$, where $i \in \{1, 2, ..., N\}, d_s \in E_d$ and $\pi_0 \in \{0, 1\}$. In the rest of this thesis, we simply use e to represent a boundary.*

Still in the example of Fig 2.2, the hybrid state $h_M = ((\pi^1_M, 1), (1, 1))$ of point M (π_M is the fractional part of h_M and π^1_M is the first value of the vector π_M) belongs to $e_1 = (B, 1, (1, 1))$, that is, the upper boundary in the second dimension (the dimension of gene B) of the discrete state 11. Since there is no other discrete state on the other side of e_1, the hybrid trajectory from h_M cannot cross e_1 and has to slide along e_1 (e_1 can be called a black wall). Boundaries like e_1, which can be reached by hybrid trajectories but cannot be crossed, are defined as *attractive boundaries*. Let us mention that the notion of black wall has been used in other hybrid systems. Generally, the behavior on black wall is not easy to define because the derivatives might be different on the different sides of a black wall. In HGRNs, by using hybrid states, a black wall is separated into two boundaries, therefore the system can have different derivatives on the different sides of the wall. There exist other methods to define behaviors of the system on a black wall [31, 32].

The hybrid state $h_P = ((\pi^1_P, 0), (0, 1))$ of point P belongs to $e_2 = (B, 0, (0, 1))$, the lower boundary in the second dimension of the discrete state 01. The hybrid trajectory from h_P reaches instantly $h_Q = ((\pi^1_Q, 1), (0, 0))$, which belongs to $e_3 = (B, 1, (0, 0))$, the upper boundary in the second dimension of discrete state 00, because the celerities on both sides allow this (instant) discrete transition. e_2 is called an *output boundary* of 01 and e_3 is called an *input boundary* of 00.

Although the hybrid trajectory in Fig 2.2 only reaches one new boundary at a time, in general a hybrid trajectory can reach several new boundaries at the same time. When a hybrid trajectory reaches several output boundaries at the same time, it can cross any of them but can only cross one boundary at a time, which causes non-deterministic behaviors, see for example Fig 2.3.

The simulation of HGRNs is presented more formally as follows.

Consider a hybrid state $h = (\pi, d_s)$ and a hybrid trajectory τ which reaches h at $t > 0$.

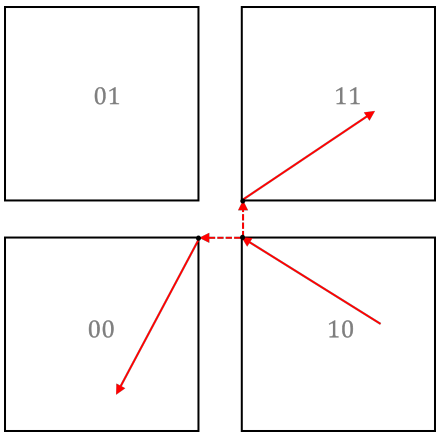


Figure 2.3 – Illustration of a non-deterministic behavior.

— If h does not belong to any boundary, then $\frac{d\tau(t)}{dt} = c_s$ (the time derivative of a hybrid state $h = (\pi, d_s)$ is defined as $\frac{dh}{dt} = \frac{d\pi}{dt}$).

— If h only belongs to one boundary e, let us consider that e is the upper boundary in i^{th} dimension (the result is easily adapted when e is the lower boundary). In case d_s^i is not the maximal discrete level of the i^{th} gene, the discrete state on the other side of e is noted as $d_{s'}$, where $d_s^k = d_{s'}^k$ for all $k \neq i$, and $d_s^i + 1 = d_{s'}^i$. There are four possible cases:

■ If $c_s^i < 0$ (c_s^i is the i^{th} value of the vector c_s), then the hybrid trajectory from the current hybrid state enters the interior of the current discrete state. e is called an *input boundary* of d_s. $\frac{d\tau(t^+)}{dt} = c_s$, $\frac{d\tau(t^-)}{dt} = c_{s'}$ and $\tau(t) = ((\pi', d_{s'}), (\pi, d_s))$, where $\pi'^k = \pi^k$ for all $k \neq i$, and $\pi'^i = 0$, which means that there is an instant transition from $(\pi', d_{s'})$ to (π, d_s) at t. For instance, the hybrid state Q in Fig 2.2 belongs to an input boundary.

■ If $c_s^i = 0$, then the hybrid trajectory from the current hybrid state will slide along the boundary e, which is then called a *neutral boundary* of d_s. $\frac{d\tau(t^+)}{dt} = \frac{d\tau(t^-)}{dt} = c_s$ and $\tau(t) = (\pi, d_s)$.

■ If $c_s^i > 0$, and either d_s^i is the maximal discrete level of the i^{th} gene, or d_s^i is not the maximal discrete level of the i^{th} gene but the i^{th} component of $c_{s'}$ is negative, then the hybrid trajectory from the current hybrid state will slide along the boundary e, which is called an *attractive boundary* of d_s. If τ reaches e at t, then: $\frac{d\tau(t^+)}{dt}^k = c_s^k$ (where $\frac{d\tau(t^+)}{dt}^k$ is the k^{th} component of the vector $\frac{d\tau(t^+)}{dt}$) for all $k \neq i$, $\frac{d\tau(t^+)}{dt}^i = 0$, $\frac{d\tau(t^-)}{dt} = c_s$ and $\tau(t) = (\pi, d_s)$. If τ reaches e at $t_0 < t$,

then: $\frac{d\tau(t)}{dt}^k = c_s^k$ for all $k \neq i$, $\frac{d\tau(t)}{dt}^i = 0$, and $\tau(t) = (\pi, d_s)$. For instance, the hybrid state M in Fig 2.2 belongs to an attractive boundary.

- If $c_s^i > 0$, d_s^i is not the maximal discrete level of the i^{th} gene, and the i^{th} component of $c_{s'}$ is positive, then the hybrid trajectory from the current hybrid state will cross instantly the boundary e and enter the discrete state $d_{s'}$. e is called an *output boundary* of d_s. $\frac{d\tau(t^+)}{dt} = c_{s'}$, $\frac{d\tau(t^-)}{dt} = c_s$ and $\tau(t) = ((\pi, d_s), (\pi', d_{s'}))$, where $\pi'^k = \pi^k$ for all $k \neq i$, and $\pi'^i = 0$. For instance, the hybrid state P in Fig 2.2 belongs to an output boundary.

— If h belongs to several boundaries, then the previous cases can be mixed:

- If in these boundaries there is no output boundary, then the hybrid trajectory from the current hybrid state will exit all input boundaries and slide along all attractive or neutral boundaries.
- If in these boundaries there is only one output boundary, then the hybrid trajectory from the current hybrid state will cross this output boundary.
- If in these boundaries there are several output boundaries, then the hybrid trajectory from the current hybrid state can cross any of them, but can only cross one boundary at one time, which causes non-deterministic behavior.

HGRNs are not yet widely used to model gene regulatory networks, but they could potentially solve some problems that are difficult to solve by using ODEs or PWAs. For example, the analysis of limit cycles in ODEs and PWAs is difficult and sometimes it is not possible to prove theoretically the existence or the stability of limit cycle, while it is feasible in HGRNs; and HGRNs can be potentially applied on large networks.

This thesis majorly focuses on HGRNs. In the following chapters, we present our works on different aspects of HGRNs, including analysis of limit cycles (Chapter 3), reachability analysis (Chapter 4), condition for sustained oscillations in the canonical repressilator (Chapter 5) and parameter identification (Chapter 6).

Chapter 3

Search and Stability Analysis of Limit Cycles

Many gene regulatory networks have periodic behavior, for instance the cell cycle or the circadian clock. Therefore, the study of formal methods to analyze limit cycles in mathematical models of gene regulatory networks is of interest. In this chapter, we study the limit cycles of HGRNs, and we present a new formal method to find all limit cycles that are simple and deterministic, and analyze their stability, that is, the ability of the model to converge back to the cycle after a small perturbation. Before this work, only 2-dimensional limit cycles of HGRNs have been studied [33]; our work fills this gap by proposing a generic approach applicable in higher dimensions. This method is based on two major ideas: Firstly, the hybrid states are abstracted to consider only their borders, in order to enumerate all simple abstract cycles containing possible concrete trajectories; secondly, a Poincaré map is used, based on the notion of transition matrix of the concrete continuous dynamics inside these abstract paths. The lower dynamical complexity of HGRN allows the application of this method on HGRN in any dimension. We apply the method on three HGRNs of negative feedback loops with 3 components, and a HGRN of the cell cycle with 5 components. This work has been presented at CMSB 2022 as regular paper [13].

3.1 Limit Cycles of HGRNs

We define that a periodic hybrid trajectory of HGRN is a limit cycle if there exists another hybrid trajectory that converges to (when time approaches positive or negative infinity) (see Fig 3.1) or reaches this periodic hybrid trajectory (see Fig 3.2).

The original definition of limit cycles, which comes from the field of non-linear dynamical systems, does not consider periodic trajectories that can be reached by other trajectories. In fact, there is no such periodic trajectories in non-linear dynamical sys-

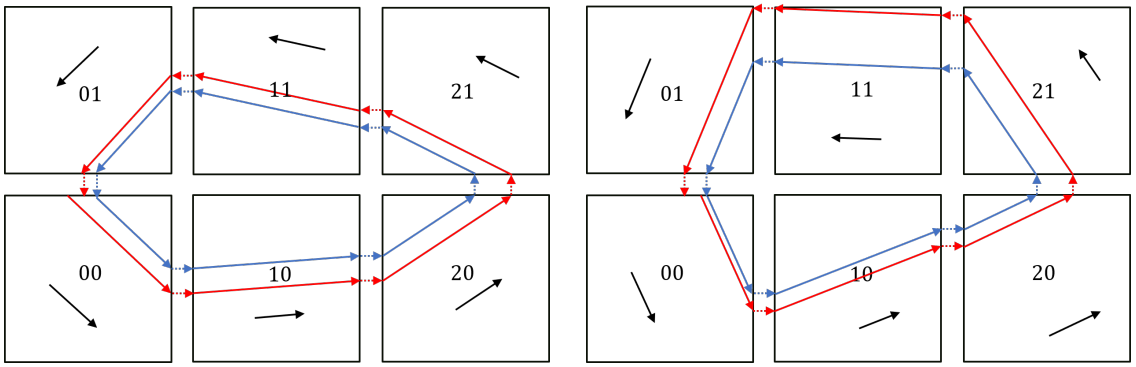


Figure 3.1 – Left: A limit cycle (the blue periodic hybrid trajectory) with another hybrid trajectory (the red hybrid trajectory) converging to it when time approaches positive infinity. Right: A limit cycle (the blue periodic hybrid trajectory) with another hybrid trajectory (the red hybrid trajectory) converging to it when time approaches negative infinity.

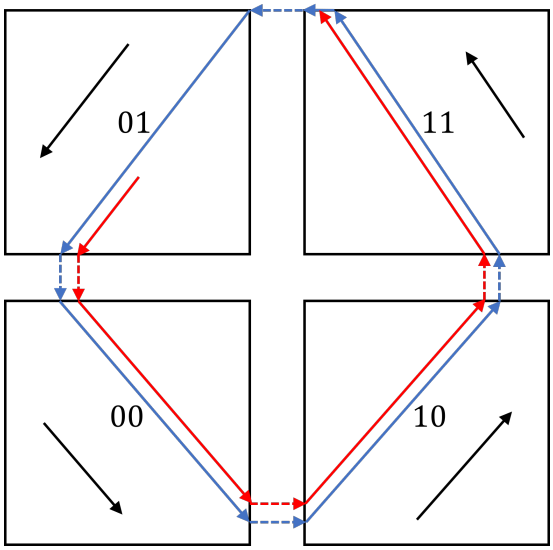


Figure 3.2 – A limit cycle (the blue periodic hybrid trajectory) with other hybrid trajectories (for instance, the red hybrid trajectory) reaching it.

tems, while the existence of sliding modes in HGRNs makes such behaviors exist. In our work, we choose to consider these periodic hybrid trajectories as limit cycles because they also show the asymptotic behaviors of the systems.

As there exist limit cycles reached by other hybrid trajectories, there exist also limit cycles such that from a hybrid state on the limit cycle, there exists a hybrid trajectory which leaves the limit cycle. Such limit cycles must contain non-deterministic hybrid

state(s). In our work, we do not consider this type of limit cycles because they are not realistic when considering real-life systems.

There exist also limit cycles of HGRN such that another hybrid trajectory converges to it with an infinite number of transitions. For example, the periodic hybrid trajectory in Fig 3.3 with hybrid states A, B, C, D, which is a limit cycle with only instant transition.

We define that a transition from h_1 to h_2 is a non-instant transition if h_1 and h_2 are two different hybrid states in the same discrete state and the hybrid trajectory from h_1 reaches a new boundary for the first time at h_2. For example the transition from N to M in Fig 2.2 is a non-instant transition. And an instant transition is a transition from a hybrid state in a given discrete state to another hybrid in a different discrete state which takes no time; for example, in Fig 2.2, the transition from P to Q is an instant transition.

We see that in Fig 3.3, there is hybrid trajectory τ converging to the limit cycle, and the intersection between this hybrid trajectory and the boundary e_1 (the lower boundary in the second dimension of the discrete state 01) are hybrid states denoted by the sequence $P_1, P_2, P_3, ...$ From P_1, this hybrid trajectory can be divided into shorter hybrid trajectories $\tau_{P_i \to P_{i+1}}$ (hybrid trajectory from P_i to P_{i+1}) by this sequence of intersection hybrid states. The duration of each shorter hybrid trajectory $\tau_{P_i \to P_{i+1}}$ is denoted by t_i. Based on the above notations, the whole duration of τ is the sum $t_\epsilon + t_1 + t_2 + t_3 + ...$ where t_ϵ is the duration of τ before reaching P_1. By the similarity between each $\tau_{P_i \to P_{i+1}}$, we can derive that this sum can be expressed by $t_\epsilon + t_1 + \alpha t_1 + \alpha^2 t_1 + \alpha^3 t_1 ...$ where α is a positive real number that is strictly less than 1. In fact, we can see that $t_1 = \frac{EF}{\|c_{00}\|_2} + \frac{GH}{\|c_{10}\|_2} + \frac{IJ}{\|c_{11}\|_2} + \frac{KP_2}{\|c_{01}\|_2}$ ($c_{00}, c_{01}, c_{10}, c_{11}$ are the celerity vectors of discrete states $00, 01, 10, 11$ and EF is the distance between E and F for instance), and $t_2 = \frac{LD}{ED}\frac{EF}{\|c_{00}\|_2} + \frac{LD}{ED}\frac{GH}{\|c_{10}\|_2} + \frac{LD}{ED}\frac{IJ}{\|c_{11}\|_2} + \frac{LD}{ED}\frac{KP_2}{\|c_{01}\|_2}$ (by the property of similarity), so $\alpha = \frac{LD}{ED}$. Since $t_1 + \alpha t_1 + \alpha^2 t_1 + \alpha^3 t_1 ...$ is the sum of a geometric sequence of which the ratio (α) is between 0 and 1, the whole duration of τ is a finite value, which means that τ converges to this limit cycle in finite time.

In fact, only limit cycles that have only instant transitions can have hybrid trajectories converging to them in finite time. All hybrid states in such limit cycle are related to the same point in Euclidean space. We say that the hybrid state (π, d_s) and the point $x \in \mathbb{R}^N$ in Euclidean space are related if $x = \pi + d_s$. Such periodic hybrid trajectory could be called a Zeno fixed point. In this chapter, we do not consider such limit cycles because they are not associated to limit cycles in Euclidean space.

There are also some periodic hybrid trajectories that are not limit cycles (see Fig 3.4).

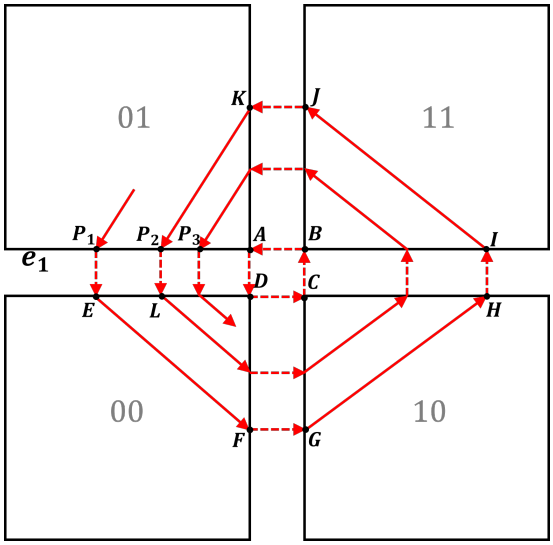


Figure 3.3 – A limit cycle with other hybrid trajectories converging to it in finite time.

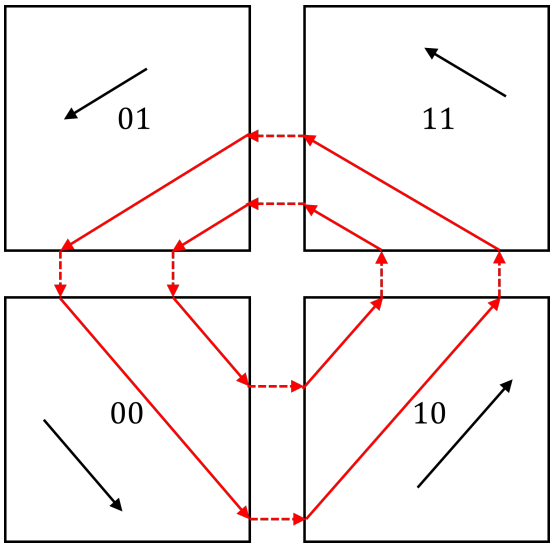


Figure 3.4 – Periodic hybrid trajectories that are not limit cycles.

3.2 Search of Periodic Hybrid Trajectories

This section presents our methods to find periodic hybrid trajectories which are potential limit cycles. In this chapter, we make two assumptions about periodic hybrid trajectories in HGRNs.

Assumption 1 *Any non-instant transition on a periodic hybrid trajectory does not reach more than one new boundary at the same time.*

Assumption 2 *For any hybrid state h on a periodic hybrid trajectory, there is at most one output boundary to which h belongs. In other words, any periodic hybrid trajectory does not have non-deterministic hybrid state.*

For Assumption 1, in real-life systems, it is indeed very unlikely for parameters to be that constrained due to measurement noise. An example of a non-instant transition reaching two new boundaries at the same time is given in Fig 2.3.

If Assumption 1 is satisfied, then there is no non-instant transition that reaches a non-deterministic hybrid state. If we further assume that a threshold of one gene only influences at most one another gene, which is a reasonable assumption for the modeling of gene regulatory networks, see for example [34], then Assumption 2 is satisfied.

The major reason why we add these two assumptions is that the cases that do not satisfy these two assumptions are very rare and more complicated to analyze. So it is not interesting to consider them in practice. However, our approaches can be generalized to these cases.

Our method to find periodic hybrid trajectories has three steps which are described in the following.

3.2.1 Abstraction of HGRN with Discrete Domains

First, the HGRN is transformed into a graph of discrete domains. A discrete domain is a new concept proposed in our work which is defined as follows.

Definition 5 (Discrete domain) *A discrete domain $\mathcal{D}(d_s, S_-, S_+)$ is a set of hybrid states inside one discrete state d_s, defined by:*

$$\mathcal{D}(d_s, S_-, S_+) = \{(\pi, d_s) \mid \forall i \in \{1, 2, ..., N\}, \pi^i \in \begin{cases} \{1\} & \text{if } i \in S_+ \\ \{0\} & \text{if } i \in S_- \\ ]0, 1[& \text{if } i \notin S_- \cup S_+ \end{cases} \}$$

where S_+ and S_- are subsets of $\{1, 2, ..., N\}$ such that $S_+ \cap S_- = \emptyset$ and $S_+ \cup S_- \neq \emptyset$. In fact, S_+ (S_-) represents the dimensions in which the upper (lower) boundaries are reached by any hybrid state $h \in \mathcal{D}(d_s, S_-, S_+)$. In the rest of this thesis, we simply use $\mathcal{D}$ to represent a discrete domain when there is no ambiguity.

In the rest of this thesis, as a notation, we add exponents to the vector representation of a discrete state to indicate which upper (lower) boundaries are reached. For instance, 11^+ denotes the discrete domain inside discrete state 11 where the upper boundary is

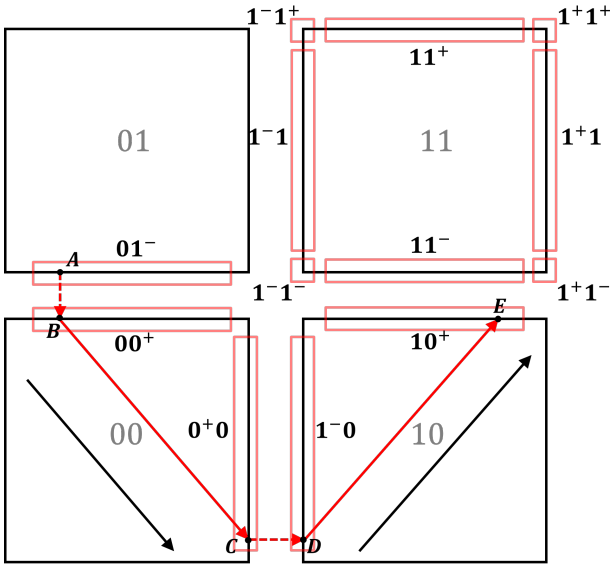


Figure 3.5 – Illustration of discrete domains and sequence of discrete domains of the HGRN in Fig 2.2.

reached for the second dimension and no boundary is reached for the first dimension, that is: $\mathcal{D}((1,1),\emptyset,\{2\}) = \{(\pi,(1,1)) \mid \pi^1 \in]0,1[\wedge \pi^2 = 1\}$ Actually, the discrete state 11 in Fig 2.2 contains 8 discrete domains: 1^-1^-, 11^-, 1^+1^-, 1^-1^+, 11^+, 1^+1^+, 1^-1 and 1^+1. These eight discrete domains are illustrated in Fig 3.5. It is worth mentioning that, in Fig 3.5, the discrete domain 1^+1^+ is illustrated by a red square, but in fact it only contains one hybrid state which is the hybrid state related to the upper-right corner of 11, and the discrete domain 11^+ is illustrated by a red rectangle representing all hybrid states on the upper boundary in the second dimension of 11 excluding these two corners.

It is then possible to build the *graph of discrete domains* of a HGRN, such as in Fig 3.6, where the nodes are the discrete domains and the edges are computed by considering only the signs of the celerities. In this graph, a discrete domain $\mathcal{D}_j$ is a successor of discrete domain $\mathcal{D}_i$ if:

— There exists an instant transition from $\mathcal{D}_i$ to $\mathcal{D}_j$, which means that at least one hybrid trajectory from $\mathcal{D}_i$ crosses a boundary and instantly reaches $\mathcal{D}_j$; see for example 0^+0 and 1^-0 in Fig 3.6 and Fig 3.5.

— Only considering the sign of celerities, it is possible that there is a hybrid trajectory which begins from $\mathcal{D}_i$ and reaches $\mathcal{D}_j$ without going through another boundary; see for example 00^+ and 0^+0 in Fig 3.6 and Fig 3.5: Since the celerity of 00 is positive in the first dimension and negative in the second, it is possible that there

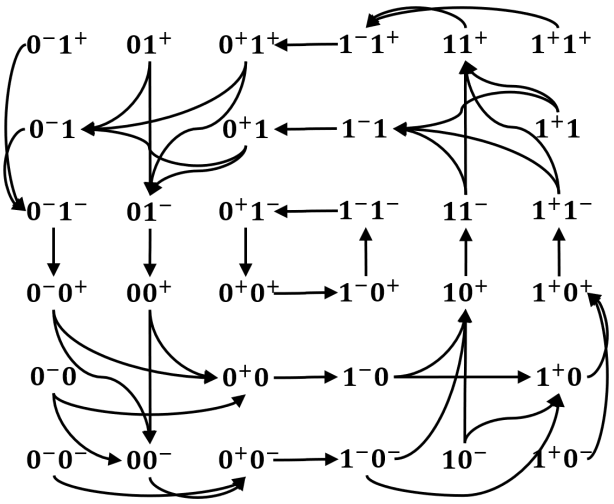


Figure 3.6 – Graph of discrete domains of the HGRN in Fig 2.2.

is a hybrid trajectory from 00^+ which reaches 0^+0. We exclude cases where two new boundaries are reached at the same time; for instance, there is no edge between 00^+ and 0^+0^-.

Need to mention that this idea of discrete abstraction is commonly used for the analysis of hybrid dynamical systems [35, 36].

3.2.2 Search of Closed Discrete Trajectories

Before giving the definition of closed discrete trajectories and the method to find them, some concepts are introduced at first.

A walk in the graph of discrete domains gives a sequence of discrete domains $\mathcal{T} = (\mathcal{D}_0, \mathcal{D}_1, ...\mathcal{D}_p)$. A hybrid trajectory is said to be *inside* such a sequence of discrete domains if it begins from the first discrete domain and reaches by order all discrete domains in the sequence. Based on this, we define two new notions on such a sequence: The *transition matrix*, which allows to compute the final hybrid state of a hybrid trajectory inside a given sequence of discrete domains, when it exists, and the *compatible zone*, which is the set of initial hybrid states so that such a hybrid trajectory exists. These two notions are presented formally as follows. Note that, there exist hybrid trajectories which are inside sequences of discrete domains that are not in the graph of discrete domains introduced in the previous subsection because the cases where two new boundaries are reached at the same time are excluded for the construction of the graph. In fact, we do not need to consider such sequences of discrete domains because Assumption 1 is made.

Definition 6 (Transition matrix) *Consider two different discrete domains $\mathcal{D}_i$ and $\mathcal{D}_j$ such that there exists a sequence of discrete domains $\mathcal{T}$ from $\mathcal{D}_i$ to $\mathcal{D}_j$. If there exists a hybrid state $h_i = (\pi_i, d_{s_i})$ in $\mathcal{D}_i$ such that from h_i there is a hybrid trajectory τ (defined on $[0, t_0]$) which is inside $\mathcal{T}$ and reaches $\mathcal{D}_j$ on $h_j = (\pi_j, d_{s_j})$ at t_0, then there exists a* transition matrix *M which describes the relation between π_i and π_j, that is: $\pi_j = s^{-1}(Ms(\pi_i))$, where s is a function that adds an extra dimension and the value in the extra dimension is always 1: $s((a_1, a_2, ..., a_N)) = (a_1, a_2, ..., a_N, 1)$. The transition matrix M only depends on $\mathcal{T}$.*

Consider Fig 3.5, the transition matrix of the sequence of discrete domains $(00^+, 0^+0)$ is $M_{(00^+,0^+0)} = \begin{bmatrix} 0 & 0 & 1 \\ -\frac{c^2_{00}}{c^1_{00}} & 1 & \frac{c^2_{00}}{c^1_{00}} \\ 0 & 0 & 1 \end{bmatrix}$ where (c^1_{00}, c^2_{00}) is the celerity vector of discrete state 00. $M_{(00^+,0^+0)}$ can describe the relation between hybrid state $h_B = ((\pi^1_B, \pi^2_B = 1), (0, 1))$ and hybrid state $h_C = ((\pi^1_C = 1, \pi^2_C), (0, 0))$:

$$\begin{pmatrix} \pi^1_C \\ \pi^2_C \\ 1 \end{pmatrix} = M_{(00^+,0^+0)} \begin{pmatrix} \pi^1_B \\ \pi^2_B \\ 1 \end{pmatrix}. \tag{3.1}$$

Similarly, for the sequence of discrete domains $(0^+0, 1^-0)$, the transition matrix is $M_{(0^+0,1^-0)} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ which describes the relation between $h_C = ((\pi^1_C = 1, \pi^2_C), (0, 0))$ and $h_D = ((\pi^1_D = 0, \pi^2_D), (1, 0))$:

$$\begin{pmatrix} \pi^1_D \\ \pi^2_D \\ 1 \end{pmatrix} = M_{(0^+0,1^-0)} \begin{pmatrix} \pi^1_C \\ \pi^2_C \\ 1 \end{pmatrix}. \tag{3.2}$$

The transition matrix can be computed by the multiplication of the transition matrices of all elementary sequences (sequences of length 2), for example $M_{(00^+,0^+0,1^-0)} = M_{(0^+0,1^-0)} M_{(00^+,0^+0)}$.

To illustrate the method to compute the transition matrix of an elementary sequence of discrete domains, the following examples are given.

First, we consider an elementary sequence in a 4-dimensional HGRN that is not instant

(meaning that all hybrid trajectories inside it are not instant): (a^+bc^-d, a^+bcd^+) where $abcd$ is a discrete state. The celerity vector of this discrete state is denoted by $(c_s^1, c_s^2, c_s^3, c_s^4)$ and we assume that $c_s^1, c_s^2, c_s^3, c_s^4 > 0$ and the upper boundary in the first dimension of $abcd$ is an attractive boundary. Any hybrid trajectory that is inside (a^+bc^-d, a^+bcd^+) can be represented by $((1, \pi_i^2, 0, \pi_i^4), abcd) \rightarrow ((1, \pi_j^2, \pi_j^3, 1), abcd)$ where there is sliding mode in the first dimension because an attractive boundary is reached. The transition matrix $M_{(a^+bc^-d, a^+bcd^+)}$ describes the relation between $(1, \pi_i^2, 0, \pi_i^4)$ and $(1, \pi_j^2, \pi_j^3, 1)$. In order to compute this relation, we compute at first the duration of this hybrid trajectory $((1, \pi_i^2, 0, \pi_i^4), abcd) \rightarrow ((1, \pi_j^2, \pi_j^3, 1), abcd)$ denoted by t_0, which is in fact the time consumed in the dimension in which new boundary is reached:

$$t_0 = \frac{1 - \pi_i^4}{c_s^4} \tag{3.3}$$

In other dimensions, the relations between $(1, \pi_i^2, 0, \pi_i^4)$ and $(1, \pi_j^2, \pi_j^3, 1)$ can be described based on t_0:

$$1 = 1 + 0 \times t_0 \tag{3.4}$$

$$\pi_j^2 = \pi_i^2 + c_s^2 \times t_0 \tag{3.5}$$

$$\pi_j^3 = 0 + c_s^3 \times t_0 \tag{3.6}$$

The temporal derivative in the first dimension is 0 (Eq (3.4)) because of the existence of the sliding mode. By reformulating all these above relations and eliminating t_0, the transition matrix $M_{(a^+bc^-d, a^+bcd^+)}$ is obtained (Eq (3.7)).

$$\begin{pmatrix} 1 \\ \pi_j^2 \\ \pi_j^3 \\ 1 \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\frac{c_S^2}{c_S^4} & \frac{c_s^2}{c_S^4} \\ 0 & 0 & 1 & -\frac{c_s^3}{c_s^4} & \frac{c_s^3}{c_S^4} \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 1 \\ \pi_i^2 \\ 0 \\ \pi_i^4 \\ 1 \end{pmatrix} \tag{3.7}$$

Second, we consider another elementary sequence in a 4-dimensional HGRN that is instant (meaning that all hybrid trajectories inside it are instant): $(a^+bcd^+, a^+bc(d+1)^-)$. Any hybrid trajectory inside $(a^+bcd^+, a^+bc(d+1)^-)$ can be represented by $((1, \pi_j^2, \pi_j^3, 1), abcd) \rightarrow ((1, \pi_j^2, \pi_j^3, 0), abc(d+1))$. Since the only difference between $(1, \pi_j^2, \pi_j^3, 1)$ and $(1, \pi_j^2, \pi_j^3, 0)$ is the value in the fourth dimension, the transition matrix $M_{(a^+bcd^+, a^+bc(d+1)^-))}$ can be obtained by modifying only one line of the identity matrix

to switch the value in the fourth dimension from 1 to 0 (Eq (3.8)).

$$\begin{pmatrix} 1 \\ \pi_j^2 \\ \pi_j^3 \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 1 \\ \pi_j^2 \\ \pi_j^3 \\ 1 \\ 1 \end{pmatrix} \tag{3.8}$$

The general method to compute the transition matrix $M_{(\mathcal{D}_i,\mathcal{D}_{i+1})}$ of any elementary sequence of discrete domains $(\mathcal{D}_i, \mathcal{D}_{i+1})$ is given as follows.

- Case 1: The transition between $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$ is instant (e.g. $(0^+0, 1^-0)$). In this case, by assuming that the boundary in dimension n_0 is crossed, $M_{(\mathcal{D}_i,\mathcal{D}_{i+1})} = (a_{kl}) \in \mathbb{R}^{(N+1)\times(N+1)}$, where
  - $a_{kl} = 1$, if $k = l$.
  - $a_{kl} = x$, if $k = n_0 \wedge l = N + 1$, where
    - $x = -1$, if the output boundary is an upper boundary.
    - $x = 1$, if the output boundary is a lower boundary.
  - $a_{kl} = 0$, otherwise.
- Case 2: The transition between $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$ is not instant (e.g. $(00^+, 0^+0)$). In this case, by assuming that a new boundary is reached in dimension n_0, $M_{(\mathcal{D}_i,\mathcal{D}_{i+1})} = (a_{kl}) \in \mathbb{R}^{(N+1)\times(N+1)}$, where
  - $a_{kl} = 1$, if $k = l \neq n_0$.
  - $a_{kl} = -\frac{c^k}{c^{n_0}}$, if $k \neq n_0 \wedge l = n_0$ and there is no sliding mode in dimension k (c is the celerity vector of the discrete state containing $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$, and c^k is the k^{th} component of c).
  - $a_{kl} = \frac{c_k}{c^{n_0}} x$, if $k \neq n_0 \wedge l = N + 1$ and there is no sliding mode in dimension k, where
    - $x = 1$, if the new reached boundary is an upper boundary.
    - $x = 0$, if the new reached boundary is a lower boundary.
  - $a_{kl} = x$, if $k = n_0 \wedge l = N + 1$. (The value of x is same as the above case.)
  - $a_{kl} = 0$, otherwise.

In case 2 of the general method to compute the transition matrix, a priori, several new boundaries can be reached, in such case, we can choose the dimension of any of them as n_0. While in this chapter, since we only consider sequences on the graph of discrete domains, there is always a unique n_0 for any elementary sequence that is not instant.

The second proposed notion associated to sequences of discrete domains is the compatible zone.

Definition 7 (Compatible zone) *Consider a sequence of discrete domains* $\mathcal{T} = (\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m)$. *The* compatible zone $\mathcal{S}$ *is the maximal subset of* $\mathcal{D}_0$ *such that from any hybrid state of* $\mathcal{S}$, *there is a hybrid trajectory that contains a sub-trajectory inside* $\mathcal{T}$. *More formally, for any hybrid state* $h \in \mathcal{S}$, *there is a hybrid trajectory* τ *from* h *and there exists* t_0 *such that the restriction of* τ *on* $[0, t_0]$ *is a hybrid trajectory inside* $\mathcal{T}$.

For a general sequence of discrete domains (a sequence of discrete domains that is not necessarily in the graph of discrete domains), the compatible zone is of the form: $\mathcal{S} = \{(\pi, d_{s_0}) \in \mathcal{D}_0 \mid W\pi > c \wedge W'\pi = c'\}$ where W, W' are matrices and c, c' are vectors. However, the compatible zone $\mathcal{S}$ of any sequence of discrete domains in the graph of discrete domains can be expressed only with linear inequalities: $\mathcal{S} = \{(\pi, d_{s_0}) \in \mathcal{D}_0 \mid W\pi > c\}$ because, inside these sequences of discrete domains, there is no hybrid trajectory that has a non-instant transition reaching two new boundaries at the same time. The idea to compute the compatible zone is based on Theorem 1.

Theorem 1 *A hybrid state* $h = (\pi, d_{s_0})$ *belongs to the compatible zone* $\mathcal{S}$ *of* $\mathcal{T} = (\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m)$ *if and only if* $(\pi, d_{s_0}) \in \mathcal{D}_0$, $(s^{-1}(M_{(\mathcal{D}_0,\mathcal{D}_1)}s(\pi)), d_{s_1}) \in \mathcal{D}_1$, $(s^{-1}(M_{(\mathcal{D}_0,\mathcal{D}_1,\mathcal{D}_2)}s(\pi)), d_{s_2}) \in \mathcal{D}_2$, ..., $(s^{-1}(M_{(\mathcal{D}_0,\mathcal{D}_1,...,\mathcal{D}_{m-1})}s(\pi)), d_{s_{m-1}}) \in \mathcal{D}_{m-1}$ *and* $(s^{-1}(M_{(\mathcal{D}_0,\mathcal{D}_1,...,\mathcal{D}_m)}s(\pi)), d_{s_m}) \in \mathcal{D}_m$, *where* $M_{(\mathcal{D}_0,\mathcal{D}_1,...,\mathcal{D}_i)}$ *is the transition matrix of* $(\mathcal{D}_0, \mathcal{D}_1, ..., \mathcal{D}_i)$ *and* $\mathcal{D}_i$ *is inside discrete state* d_{s_i} *(*$i \in \{0, 1, ..., m\}$*).*

Proof: Proof of sufficient condition: We can easily see that if $h = (\pi, d_{s_0})$ belongs to the compatible zone $\mathcal{S}$ of $\mathcal{T} = (\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m)$, then $\forall i \in \{1, 2, ..., m\}$, h also belongs to the compatible zone of $(\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_i)$, so $(s^{-1}(M_{(\mathcal{D}_0,\mathcal{D}_1,...,\mathcal{D}_i)}s(\pi)), d_{s_i}) \in \mathcal{D}_i$.

Proof of necessary condition: By induction. Consider a sequence of discrete domains of length 2: $(\mathcal{D}_0, \mathcal{D}_1)$, it is evident that h belongs to the compatible zone of $(\mathcal{D}_0, \mathcal{D}_1)$ if $(s^{-1}(M_{(\mathcal{D}_0,\mathcal{D}_1)}s(\pi)), d_{s_1}) \in \mathcal{D}_1$. Now suppose that it is true for any sequence of discrete domains of length $k + 1$, and consider a sequence of discrete domains of length $k + 2$: $(\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_{k+1})$. Since it is true for a sequence of discrete domains of length $k+1$, h belongs to the compatible zone of $(\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_k)$, so the hybrid trajectory from h will stay inside $(\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_k)$ and will reach $\mathcal{D}_k$ at $h_k = (s^{-1}(M_{(\mathcal{D}_0,\mathcal{D}_1,...,\mathcal{D}_k)}s(\pi)), d_{s_k})$. Let $h_k = (\pi_k, d_{s_k})$. We can easily see that $s^{-1}(M_{(\mathcal{D}_0,\mathcal{D}_1,...,\mathcal{D}_{k+1})}s(\pi)) = s^{-1}(M_{(\mathcal{D}_k,\mathcal{D}_{k+1})}s(\pi_k))$. Since $(s^{-1}(M_{(\mathcal{D}_0,\mathcal{D}_1,...,\mathcal{D}_{k+1})}s(\pi)), d_{s_{k+1}}) \in \mathcal{D}_{k+1}$, we have $(s^{-1}(M_{(\mathcal{D}_k,\mathcal{D}_{k+1})}s(\pi_k)), d_{s_{k+1}}) \in$

$\mathcal{D}_{k+1}$. So h_k belongs to the compatible zone of $(\mathcal{D}_k, \mathcal{D}_{k+1})$. Therefore, h belongs to the compatible zone of $(\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_{k+1})$.

□

Thus, all hybrid states that satisfy the constraints in Theorem 1 constitute the compatible zone. And we can see that the computation of compatible zone replies on the computation of transition matrix.

Now we give the definition of *closed discrete discrete trajectories*. A sequence of discrete domains $\mathcal{T}$ is called a *discrete trajectory* if the compatible zone of $\mathcal{T}$ is not empty, in other words, there exists at least one hybrid trajectory inside $\mathcal{T}$. A discrete trajectory $\mathcal{T} = (\mathcal{D}_1, \mathcal{D}_2, ...\mathcal{D}_m)$ is said *closed* if $\mathcal{D}_1 = \mathcal{D}_m$. For example, the sequence of discrete domains $(01^-, 00^+, 0^+0, 1^-0, 10^+)$ in Fig 3.5 and Fig 3.6 is a discrete trajectory because its compatible zone is not empty as there exists at least one hybrid trajectory inside this sequence (e.g. the hybrid trajectory from A to E in Fig 3.5).

In order to find closed discrete trajectories, we use a depth first algorithm. For this, we rely on the notion of *Poincaré section* (more explanations about Poincaré section will be given in the next subsection, here we can simply consider that a Poincaré section is a boundary of a discrete state). We first choose one or several input boundaries of discrete states as Poincaré sections, more precisely an input boundary is chosen if it is crossed by a cycle of discrete states, and then on each discrete domain on these Poincaré sections, we apply this depth first algorithm. In each step of this depth first algorithm, the compatible zone is computed and the search continues if the compatible zone is not empty and the current path does not reach the same discrete domain for more than one time and does not return to the initial discrete state. This algorithm finds all discrete trajectories that begin from a discrete domain and return to the initial discrete state without crossing the same discrete state more than once. An execution of this algorithm on discrete domain 00^+ of the HGRN in Fig 2.2 is illustrated in Fig 3.7. Among these discrete trajectories, we can easily find the closed ones. Note that in Fig 3.7, all discrete trajectories from 00^+ that return to 00 happen to be closed but generally we can find discrete trajectories that are not closed.

Consider the HGRN in Fig 2.2, we can easily see that there is only one cycle of discrete states in this system, which is $00 \to 10 \to 11 \to 01 \to 00$. Therefore, for this system, we only need one Poincaré section and any boundary in this cycle can take this role. Let us choose for instance the input boundary of discrete state 00 (the boundary between 01 and 00) as Poincaré section, that is, the union of the three discrete domains

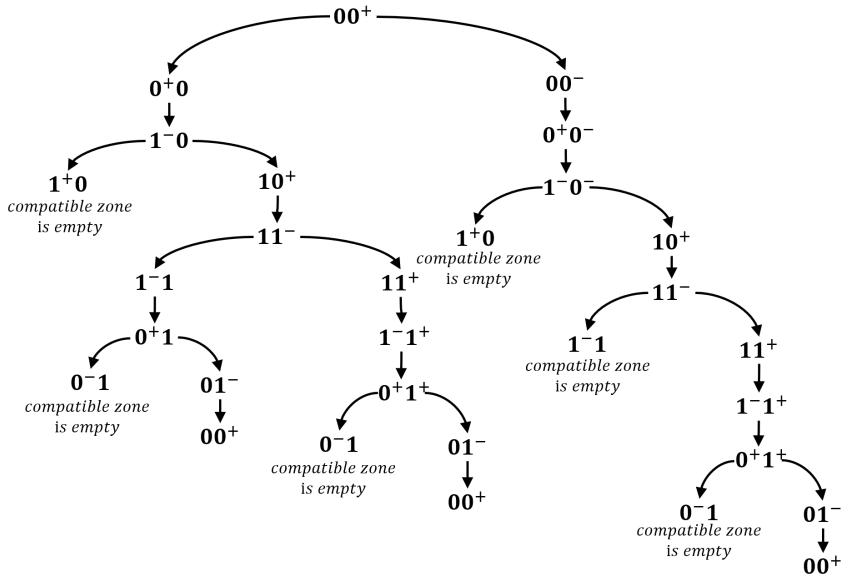


Figure 3.7 – Illustration of the depth first algorithm on discrete domain $(0, 0^+)$.

0^-0^+, 00^+ and 0^+0^+. We thus apply the depth first algorithm on each of these three discrete domains. As a result, we can find 5 discrete trajectories that begin from the Poincaré section and return to the initial discrete state:

$1 : 0^-0^+ \to 00^- \to 0^+0^- \to 1^-0^- \to 10^+ \to 11^- \to 11^+ \to 1^-1^+ \to 0^+1^+ \to 01^- \to 00^+$

$2 : 00^+ \to 0^+0 \to 1^-0 \to 10^+ \to 11^- \to 1^-1 \to 0^+1 \to 01^- \to 00^+$

$3 : 00^+ \to 0^+0 \to 1^-0 \to 10^+ \to 11^- \to 11^+ \to 1^-1^+ \to 0^+1^+ \to 01^- \to 00^+$

$4 : 00^+ \to 00^- \to 0^+0^- \to 1^-0^- \to 10^+ \to 11^- \to 11^+ \to 1^-1^+ \to 0^+1^+ \to 01^- \to 00^+$

$5 : 0^+0^+ \to 1^-0^+ \to 1^-1^- \to 0^+1^- \to 0^+0^+$

Examples of hybrid trajectories inside each of these 5 discrete trajectories are shown in Fig 3.8. We note that there always exists at least one hybrid trajectory inside a discrete trajectory since, by definition, its compatible zone is not empty. Among the 5 discrete trajectories above, only the first one is not closed.

3.2.3 Identification of Periodic Hybrid Trajectories with Poincaré Map

Consider a closed discrete trajectory $\mathcal{T} = (\mathcal{D}_0, \mathcal{D}_1, ...\mathcal{D}_m, \mathcal{D}_0)$. We define that a periodic hybrid trajectory τ is inside $\mathcal{T}$ if τ begins from a hybrid state $h \in \mathcal{D}_0$, reaches by order all discrete domains of $\mathcal{T}$ and finally reaches back the hybrid state h. To check if there

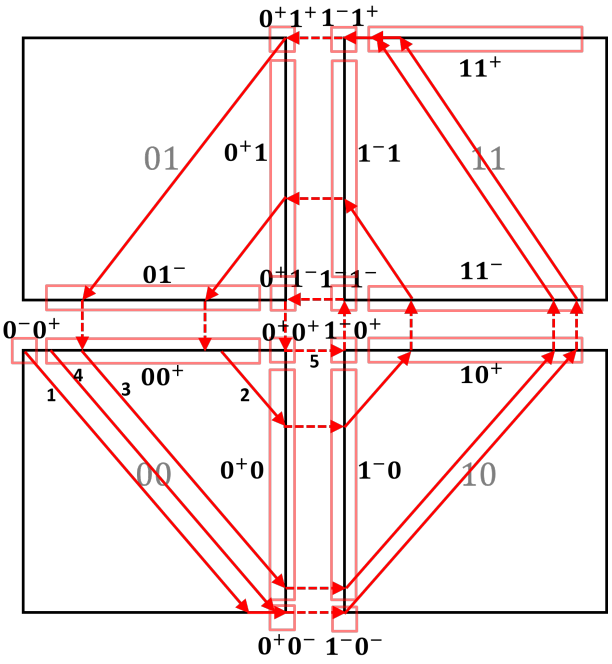


Figure 3.8 – Examples of hybrid trajectories inside different discrete trajectories of the HGRN in Fig 2.2.

is a periodic hybrid trajectory inside $\mathcal{T} = (\mathcal{D}_0, \mathcal{D}_1, ...\mathcal{D}_m, \mathcal{D}_0)$, we only need to verify the two following properties:

— $\exists(\pi_0, d_{s_0}) \in \mathcal{D}_0$ such that $s^{-1}(M_{\mathcal{T}} s(\pi_0)) = \pi_0$, and

— (π_0, d_{s_0}) belongs to the compatible zone of $\mathcal{T}$.

Then (π_0, d_{s_0}) is called a fixed point of $\mathcal{T}$, the function $s^{-1}(M_{\mathcal{T}} s())$ is the Poincaré map of the periodic hybrid trajectory that corresponds to this fixed point and $\mathcal{D}_0$ (or the boundary that contains $\mathcal{D}_0$) is the associated Poincaré section.

Under Assumption 1, any periodic hybrid trajectory of the system must be inside one of the closed discrete trajectories found by the depth first algorithm. Meanwhile, if a periodic hybrid trajectory reaches more than one new boundary at the same time, then it is not inside any closed discrete trajectories found by the algorithm.

Among the five closed discrete trajectories in the HGRN in Fig 2.2, we find only one periodic hybrid trajectory of interest, inside the third closed discrete trajectory; it is the hybrid trajectory labeled "3" in Fig 3.8. For this periodic hybrid trajectory, $\pi_0 = (0.222, 1)$, $M_{\mathcal{T}} = \begin{bmatrix} 0 & 0 & 0.222 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$, and the compatible zone is

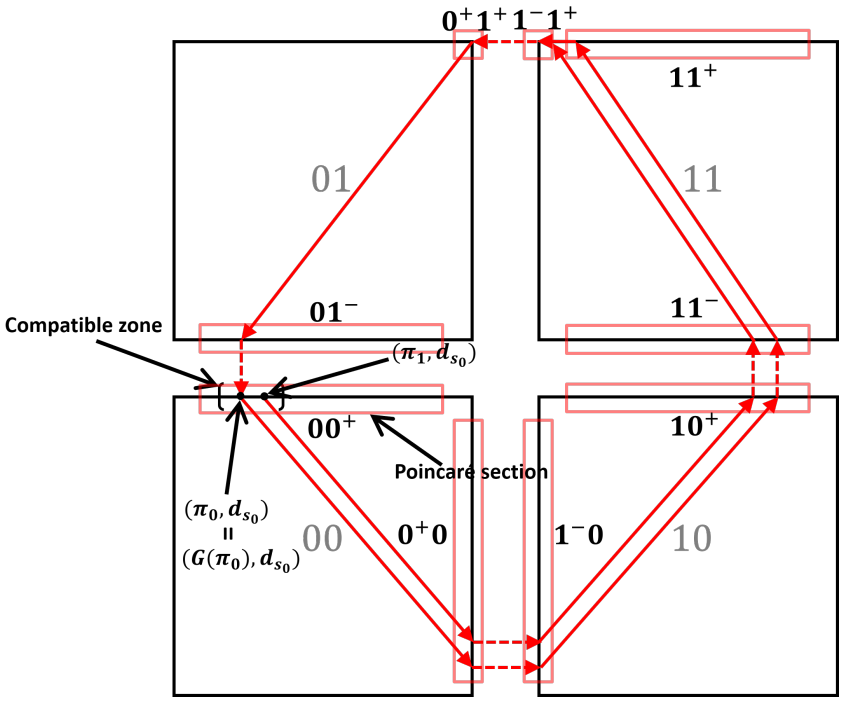


Figure 3.9 – Illustration of the Poincaré map of a periodic hybrid trajectory of the HGRN in Fig 2.2, where $G() = s^{-1}(M_{\mathcal{T}} s())$ is the Poincaré map.

$\{(\pi, (0,0)) \mid \pi^2 = 1, \pi^1 \in]0.1428, 0.3469[\}$. This periodic hybrid trajectory and the associated Poincaré map/section are illustrated in Fig 3.9. In this figure, the hybrid state (π_0, d_{s_0}) is the fixed point of this Poincaré map that belongs to the compatible zone, so this hybrid state is the intersection between this periodic hybrid trajectory and the Poincaré section. Generally in the field of non-linear dynamics, a Poincaré map describes the dynamics of the intersection point between a periodic trajectory and a lower dimension subspace (Poincaré section). In HGRNs, a Poincaré map not only describes the dynamics of a periodic hybrid trajectory but also describes the dynamics of the intersection hybrid states between all hybrid trajectories from the compatible zone and the Poincaré section. For example, in Fig 3.9, for any hybrid state in the compatible zone, noted by (π_1, d_{s_0}), the hybrid trajectory from this hybrid state returns to the Poincaré section at $(G(\pi_1), d_{s_0})$, which in this particular case happens to be the fixed point of the Poincaré map.

Actually, there is another periodic hybrid trajectory inside the fifth closed discrete trajectory; it is the periodic hybrid trajectory labeled "5" in Fig 3.8. This periodic hybrid trajectory only contains instant transitions. As stated before, our analysis method of limit cycles does not consider this type of periodic hybrid trajectories.

The Poincaré map has been widely used in the literature to study limit cycles of hybrid systems [15–19, 37–40]. The dynamical simplicity of HGRNs and the existence of sliding modes make the form of Poincaré maps of HGRNs particular: A Poincaré map is a piecewise affine function on a Poincaré section.

3.3 Stability Analysis of Limit Cycles

In order to define the stability of limit cycles, we give the definition of the neighborhood in the same discrete state at first.

Definition 8 (Neighborhood in the same discrete state) *The* neighborhood in the same discrete state *of a hybrid state* $h = (\pi_0, d_s)$ *is a set of hybrid states defined as:* $N_d(h, r) = \left\{(\pi, d_s) \mid d(\pi, \pi_0) < r, \pi \in [0,1]^N\right\}$, *with* $r > 0$ *the radius of this neighborhood, and* d *the maximum norm between vectors:* $d(\pi, \pi_0) = \max_{i \in \{1,2,\ldots,N\}} \mid \pi^i - \pi_0^i \mid$.

Based on the definition of neighborhood in the same discrete state, the stability of limit cycle is defined as follows.

Definition 9 (Stability of limit cycles in HGRNs) *A limit cycle* τ *of HGRN is stable if, for any hybrid state* h *on* τ, *there exists a neighborhood in the same discrete state of radius* r *such that any hybrid trajectory* τ_0 *that begins from this neighborhood* $N_d(h, r)$ *satisfies:* $\lim_{t \to \infty} (Dis_{min}(\tau_0(t), \tau)) = 0$ *where* $Dis_{min}(h', \tau)$ *is defined as* $Dis_{min}(h', \tau) = \min_{h_0 \in \tau} d(x(h'), x(h_0))$, *with* $h' \in E_h$, $x(.)$ *the sum (dimension by dimension) of the fractional part and the discrete state of a hybrid state, and* $d(v_1, v_2) = \max_{1 \leq i \leq n} |v_1^i - v_2^i|$ *(v_1, v_2 are two vectors of length n), otherwise it is unstable.*

It is noteworthy that in most cases, a value of t high enough is sufficient to obtain $Dis_{min}(\tau_0(t), \tau) = 0$, without needing a limit computation. This definition of stability is illustrated in Fig 3.10 by a 2-dimensional stable limit cycle.

3.3.1 Continuity in the Neighborhood of Periodic Hybrid Trajectories

We call *neighborhood of a hybrid trajectory* a union of neighborhoods in the same discrete state of all the hybrid states in this hybrid trajectory. A periodic hybrid trajectory is said to respect the *continuity of neighborhood* if there exists a neighborhood of this

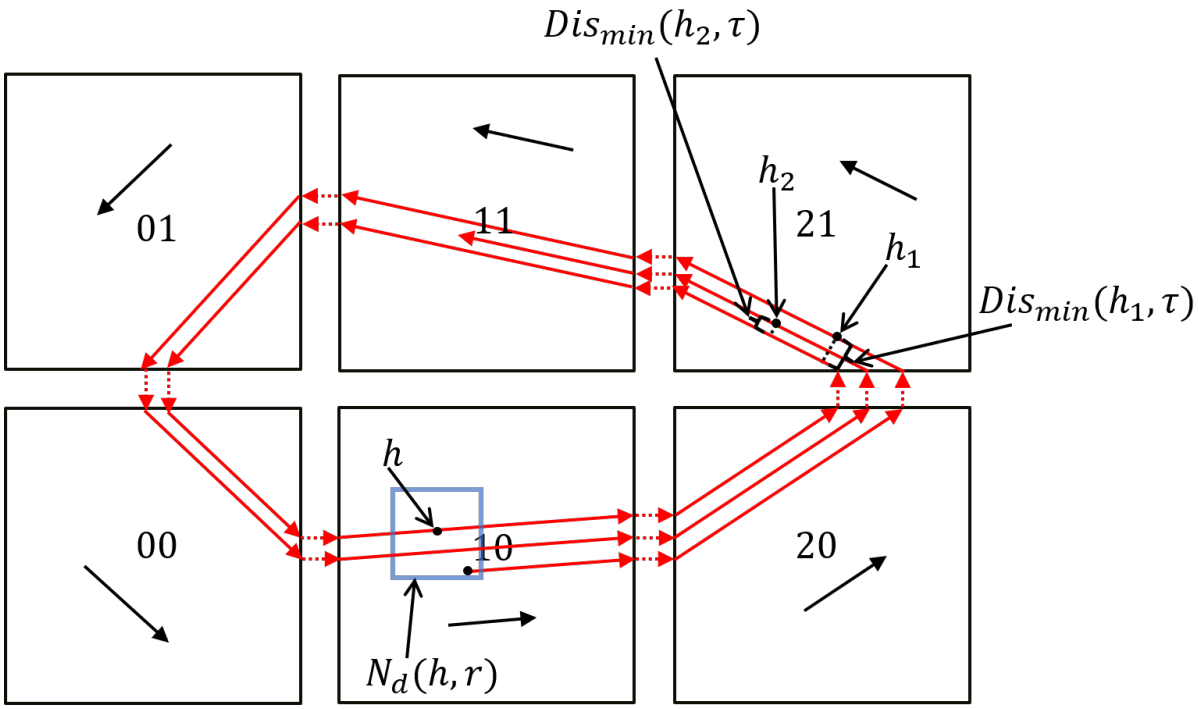


Figure 3.10 – Illustration of Definition 9 (Stability of limit cycles in HGRNs).

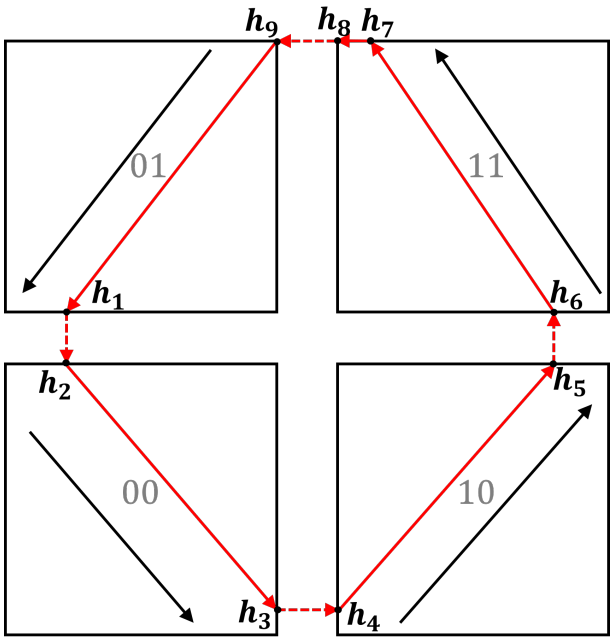


Figure 3.11 – The sequence of turning states of a periodic hybrid trajectory τ: $T_s(\tau) = (h_1, h_2, ..., h_9)$.

hybrid trajectory that is small enough so that all hybrid trajectories starting from this neighborhood remain in this neighborhood. The continuity of neighborhood is formally defined after the following definitions.

Definition 10 (Neighborhood in the same discrete domain) *The neighborhood in the same discrete domain of a hybrid state $h = (\pi_0, d_s)$ is a set of hybrid states defined as: $N_{\mathcal{D}}(h, r) = \{(\pi, d_s) \in \mathcal{D}_i \mid d(\pi, \pi_0) < r\}$, where $\mathcal{D}_i$ is the discrete domain which includes h.*

Definition 11 (Turning states of a periodic hybrid trajectory) *Consider a periodic hybrid trajectory τ, we can find a sequence of hybrid states on τ: $(h_1, h_2, ..., h_m)$, such that each h_i represents a hybrid state when τ reaches a new discrete domain and there is a transition from h_i to h_{i+1} where $i \in \{1, 2, ..., m-1\}$ and from h_m to h_1. All hybrid states in this sequence are the turning states of τ and this sequence is called the sequence of turning states of τ denoted by $T_s(\tau)$.*

The sequence of turning states of a periodic hybrid trajectory is given in Fig 3.11.

Definition 12 (Continuity of neighborhood of a periodic hybrid trajectory) *The neighborhood of a periodic hybrid trajectory τ is continuous if for any neighborhood*

in the same discrete domain of a turning state h_0 of τ: $N_{\mathcal{D}}(h_0, r_0)$ and for any hybrid state h_1 on τ, there exists a small neighborhood in the same discrete state of h_1: $N_d(h_1, r_1)$, such that all hybrid trajectories from $N_d(h_1, r_1)$ will reach $N_{\mathcal{D}}(h_0, r_0)$ in finite time.

When a periodic hybrid trajectory does not have continuity of neighborhood, some hybrid trajectories in the neighborhood may undergo a "disruption" by touching another boundary and thus follow another sequence of discrete states.

Without Assumption 1 and Assumption 2, some neighborhoods of a periodic hybrid trajectory might not respect this continuity, no matter how small they are. For example, consider a periodic hybrid trajectory that contains a transition $((1,1),(a,b)) \rightarrow ((0,1),(a+1,b))$, where the upper boundaries in the first and second dimensions of discrete state *ab* are both output boundaries. In the neighborhood of $((1,1),(a,b))$, no matter how small it is, we can always find a hybrid state from which the hybrid trajectory reaches the boundary in the second dimension at first, and as it will reach a different discrete state, it might never return to the neighborhood of this periodic hybrid trajectory.

We claim that Assumption 1 and Assumption 2 together constitute a sufficient condition for the continuity of neighborhood of any periodic hybrid trajectories in HGRNs. To prove this, we give some definitions and propositions as follows.

Proposition 1 *Consider a periodic hybrid trajectory τ with the sequence of turning states $T_s(\tau) = (h_1, h_2, ..., h_m)$. For any two adjacent turning states h_i and h_j ($j = i+1$ if $i \in \{1, 2, ..., m-1\}$, $j = 1$ if $i = m$), for any neighborhood in the same discrete domain of h_j: $N_{\mathcal{D}}(h_j, r_j)$, if Assumption 1 and Assumption 2 are satisfied, there exists a neighborhood in the same discrete domain of h_i: $N_{\mathcal{D}}(h_i, r_i)$, such that any hybrid trajectory from $N_{\mathcal{D}}(h_i, r_i)$ reaches directly $N_{\mathcal{D}}(h_j, r_j)$ (we say that a hybrid trajectory reaches directly $N_{\mathcal{D}}(h_j, r_j)$ if it reaches a new boundary for the first time at a hybrid state that belongs to $N_{\mathcal{D}}(h_j, r_j)$).*

Proposition 1 is illustrated in Fig 3.12.

Proof: If the transition between h_i and h_j is an instant transition, then h_i and h_j are related to the same point in Euclidean space, with Assumption 2, we can easily prove this proposition by choosing the same $N_{\mathcal{D}}(h_i, r_i)$ as $N_{\mathcal{D}}(h_j, r_j)$. In fact, without Assumption 2, a priori, h_i can be non-deterministic, in such case we do not have Proposition 1.

If the transition is not instant, without loss of generality, consider a 4-dimensional example: $h_i = ((1, 0, \pi_i^3, \pi_i^4), d_s)$ and $h_j = ((1, \pi_j^2, 1, \pi_j^4), d_s)$. We suppose that the celerity

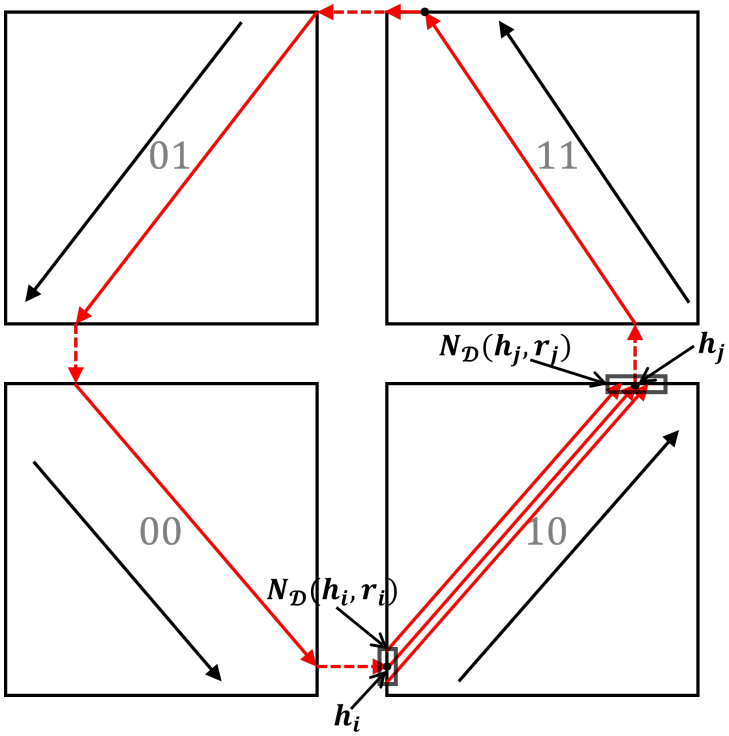


Figure 3.12 – Illustration of Proposition 1. h_i and h_j are two adjacent turning states on a periodic hybrid trajectory. Black boxes represent neighborhoods in the same discrete domain of h_i/h_j that correspond to Proposition 1.

related to this transition is $(c_s^1, c_s^2, c_s^3, c_s^4)$ and we can see that there is a sliding mode in the first dimension. The duration of this transition is noted as t_0, we have:

$$\begin{aligned} 1 &= 1 + t_0 \times 0 \\ \pi_j^2 &= 0 + t_0 \times c_s^2 \\ \pi_j^4 &= \pi_i^4 + t_0 \times c_s^4 \end{aligned} \tag{3.9}$$

where

$$t_0 = \frac{1 - \pi_i^3}{c_s^3}. \tag{3.10}$$

Any hybrid state from the neighborhood in the same discrete domain of h_i can be represented by $h_i' = ((1, 0, \pi_i^3 + \delta_3, \pi_i^4 + \delta_4), d_s)$. Suppose that the hybrid trajectory from h_i' still reaches directly the discrete domain which includes h_j at hybrid state $h_j' = ((1, \pi_j'^2, 1, \pi_j'^4), d_s)$. The duration of this new transition becomes:

$$t_1 = \frac{1 - \pi_i^3 - \delta_3}{c_s^3} = \frac{1 - \pi_i^3}{c_s^3} - \frac{\delta_3}{c_s^3} = t_0 - \frac{\delta_3}{c_s^3}. \tag{3.11}$$

And

$$\begin{aligned} 1 &= 1 + t_1 \times 0 \\ \pi_j'^2 &= 0 + t_1 \times c_s^2 \\ \pi_j'^4 &= \pi_i^4 + \delta_4 + t_1 \times c_s^4 \end{aligned} \tag{3.12}$$

By combining Eq (3.9), Eq (3.11) and Eq (3.12), we have:

$$\begin{aligned} \pi_j'^2 &= \pi_j^2 - \frac{\delta_3}{c_s^3} \times c_s^2 \\ \pi_j'^4 &= \pi_i^4 + \delta_4 - \frac{\delta_3}{c_s^3} \times c_s^4 \end{aligned} \tag{3.13}$$

To ensure that h_j' is included in $N_{\mathcal{D}}(h_j, r_j)$, firstly these constraints must be satisfied:

$$\begin{aligned} \mid \frac{\delta_3}{c_s^3} \times c_s^2 \mid &< r_j \\ \mid \delta_4 - \frac{\delta_3}{c_s^3} \times c_s^4 \mid &< r_j \end{aligned} \tag{3.14}$$

Secondly, we need to ensure that the hybrid trajectory from h'_i reaches the boundary in the third dimension at first, so these constraints are also needed to be satisfied:

$$\begin{aligned} \frac{1-\pi_i^3-\delta_3}{c_s^3} &< \frac{1-\pi_i^4-\delta_4}{c_s^4} \\ \frac{1-\pi_i^3-\delta_3}{c_s^3} &< \frac{1-0}{c_s^2} \end{aligned} \tag{3.15}$$

Here we suppose that c_s^4 is positive. We can derive from Eq (3.14) and Eq (3.15) by using triangle inequality that a sufficient condition, for the constraints in Eq (3.14) and Eq (3.15) to be satisfied, is:

$$\begin{aligned} \mid \delta_3 \mid &< r_j \frac{\mid c_s^3 \mid}{\mid c_s^2 \mid} \\ \mid \delta_3 \mid + \mid \delta_4 \mid &< \frac{r_j}{\max(1, \frac{|c_s^4|}{|c_s^3|})} \\ \mid \delta_3 \mid + \mid \delta_4 \mid &< \frac{\frac{1-\pi_i^4}{c_s^4} - \frac{1-\pi_i^3}{c_s^3}}{\max(\frac{1}{|c_s^3|}, \frac{1}{|c_s^4|})} \\ \mid \delta_3 \mid &< \mid c_s^3 \mid (\frac{1}{c_s^2} - \frac{1-\pi_i^3}{c_s^3}) \end{aligned} \tag{3.16}$$

From any hybrid state $h'_i = ((1, 0, \pi_i^3 + \delta_3, \pi_i^4 + \delta_4), d_s)$ that satisfies the above constraints (Eq (3.16)), the hybrid trajectory reaches directly $N_{\mathcal{D}}(h_j, r_j)$. So any hybrid trajectories from $N_{\mathcal{D}}(h_i, r_i)$ reaches directly $N_{\mathcal{D}}(h_j, r_j)$ if r_i satisfies:

$$r_i < \min(r_j \frac{\mid c_s^3 \mid}{\mid c_s^2 \mid}, \frac{1}{2} \frac{r_j}{\max(1, \frac{|c_s^4|}{|c_s^3|})}, \frac{1}{2} \frac{\frac{1-\pi_i^4}{c_s^4} - \frac{1-\pi_i^3}{c_s^3}}{\max(\frac{1}{|c_s^3|}, \frac{1}{|c_s^4|})}, \mid c_s^3 \mid (\frac{1}{c_s^2} - \frac{1-\pi_i^3}{c_s^3})) \tag{3.17}$$

In the above demonstration, we consider a transition of a HGRN in four dimensions and propose a method to find a $N_{\mathcal{D}}(h_i, r_i)$ that satisfies the Proposition 1. The same method can be used to find such a $N_{\mathcal{D}}(h_i, r_i)$ in any HGRN in N dimension (the only difference is that there will be more inequalities in Eq (3.14) and Eq (3.15)), if Assumption 1 and Assumption 2 are satisfied. In fact, Assumption 1 ensures that the constraints on these δ can be represented by some inequalities of the sum of absolute values of these δ (like Eq (3.16)). If Assumption 1 is not satisfied, then we need stronger constraints on these δ represented by linear equations. □

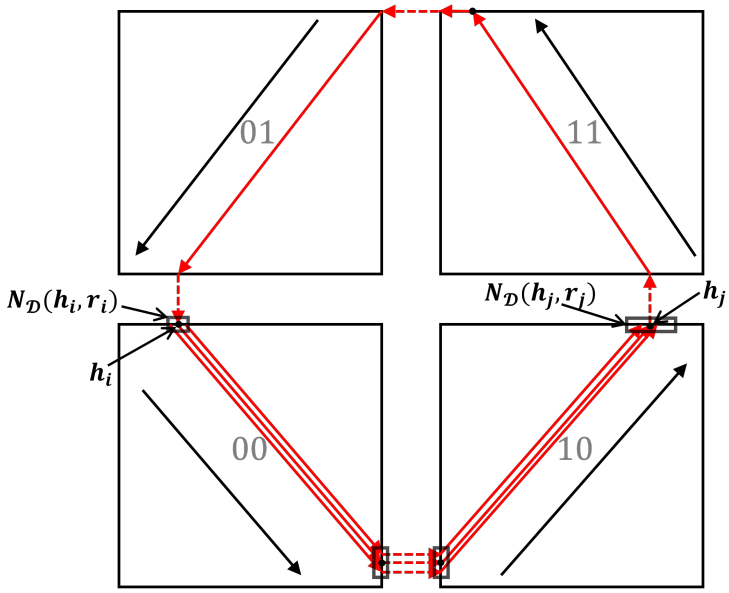


Figure 3.13 – Illustration of Proposition 2. h_i and h_j are two turning states on a periodic hybrid trajectory. Black boxes represent neighborhoods in the same discrete domain of h_i/h_j that correspond to Proposition 2.

Proposition 2 *Consider a periodic hybrid trajectory τ with the sequence of turning states $T_s(\tau) = (h_1, h_2, ..., h_m)$. For any two turning states h_i and h_j, for any neighborhood in the same discrete domain of h_j: $N_{\mathcal{D}}(h_j, r_j)$, if Assumption 1 and Assumption 2 are satisfied, there exists a neighborhood in the same discrete domain of h_i: $N_{\mathcal{D}}(h_i, r_i)$, such that all hybrid trajectories from $N_{\mathcal{D}}(h_i, r_i)$ reach by order all discrete domains between h_i and h_j (the discrete domains containing respectively h_i,h_{i+1},h_{i+2},...,h_{j-1},h_j) and finally reach $N_{\mathcal{D}}(h_j, r_j)$.*

Proposition 2 is illustrated in Fig 3.13.

Proof: Consider two turning states h_i and h_j, since τ is a periodic hybrid trajectory, we can find turning states between h_i and h_j, noted as h_{i+1}, h_{i+2}, ..., h_{j-1}, such that there are non-instant or instant transitions between h_k and h_{k+1}, $\forall k \in \{i, i+1, ..., j-1\}$. (If $j < i$, we can take h_i as the first turning state in the series of turning states of τ and renumber h_j). Consider a neighborhood in the same discrete domain of h_j: $N_{\mathcal{D}}(h_j, r_j)$, according to Proposition 1 we can find a neighborhood in the same discrete domain of h_{j-1}: $N_{\mathcal{D}}(h_{j-1}, r_{j-1})$, such that all hybrid trajectories from $N_{\mathcal{D}}(h_{j-1}, r_{j-1})$ reach directly $N_{\mathcal{D}}(h_j, r_j)$. Then, in the same way, we can find a neighborhood in the same discrete domain

of h_{j-2}: $N_{\mathcal{D}}(h_{j-2}, r_{j-2})$, such that all hybrid trajectories from $N_{\mathcal{D}}(h_{j-2}, r_{j-2})$ reach directly $N_{\mathcal{D}}(h_{j-1}, r_{j-1})$. Using this backward method, we can finally find a neighborhood in the same discrete domain of h_i: $N_{\mathcal{D}}(h_i, r_i)$, such that all hybrid trajectories from $N_{\mathcal{D}}(h_i, r_i)$ firstly reach $N_{\mathcal{D}}(h_{i+1}, r_{i+1})$, then reach $N_{\mathcal{D}}(h_{i+2}, r_{i+2})$, ..., and finally reach $N_{\mathcal{D}}(h_j, r_j)$. □

Proposition 3 *Consider a periodic hybrid trajectory τ with the sequence of turning states $T_s(\tau) = (h_1, h_2, ..., h_m)$. For any two adjacent turning states h_i and h_j ($j = i + 1$ if $i \in \{1, 2, ..., m-1\}$, $j = 1$ if $i = m$) such that the transition from h_i to h_j is a non-instant transition (the hybrid trajectory from h_i to h_j is denoted by τ_i), for any neighborhood in the same discrete domain of h_j: $N_{\mathcal{D}}(h_j, r_j)$ and for any hybrid state h_k on τ_i (including h_i but not including h_j), if Assumption 1 is satisfied, there exists a neighborhood in the same discrete state of h_k: $N_d(h_k, r_k)$, such that all hybrid trajectories from $N_d(h_k, r_k)$ reach $N_{\mathcal{D}}(h_j, r_j)$.*

Proposition 3 is illustrated in Fig 3.14.

Proof: Consider two adjacent turning states: $h_i = ((1, 0, \pi_i^3, \pi_i^4), d_s)$ and $h_j = ((1, \pi_j^2, 1, \pi_j^4), d_s)$ (same example used in the proof of Proposition 1) and the neighborhood in the same discrete domain of h_j: $N_{\mathcal{D}}(h_j, r_j)$.
Firstly, we consider h_i. Any hybrid state in a neighborhood of the same discrete state of h_i can be represented by $h_i' = ((1 + \delta_1, 0 + \delta_2, \pi_i^3 + \delta_3, \pi_i^4 + \delta_4), d_s)$ (we can see that δ_1 must be negative and δ_2 must be positive). We suppose that the hybrid trajectory from h_i' firstly reaches the upper boundary in the first dimension (an attractive boundary) at the hybrid state $h_i'' = ((1, 0 + \delta_2 - \frac{\delta_1 c_s^2}{c_s^1}, \pi_i^3 + \delta_3 - \frac{\delta_1 c_s^3}{c_s^1}, \pi_i^4 + \delta_4 - \frac{\delta_1 c_s^4}{c_s^1}), d_s)$. To ensure that the hybrid trajectory from h_i'' reaches $N_{\mathcal{D}}(h_j, r_j)$, we can use the same method as in the proof of Proposition 1 to get some inequalities of the sum of absolute values of these δ (like Eq (3.16)) and we call these linear inequalities Constraint 1. To ensure that the hybrid trajectory from h_i' firstly reaches the upper boundary in the first dimension, the following inequalities must be satisfied:

$$\begin{aligned} \frac{-\delta_1}{c_s^1} &< \frac{1 - \delta_2}{c_s^2} \\ \frac{-\delta_1}{c_s^1} &< \frac{1 - \delta_3 - \pi_i^3}{c_s^3} \\ \frac{-\delta_1}{c_s^1} &< \frac{1 - \delta_4 - \pi_i^4}{c_s^4} \end{aligned} \tag{3.18}$$

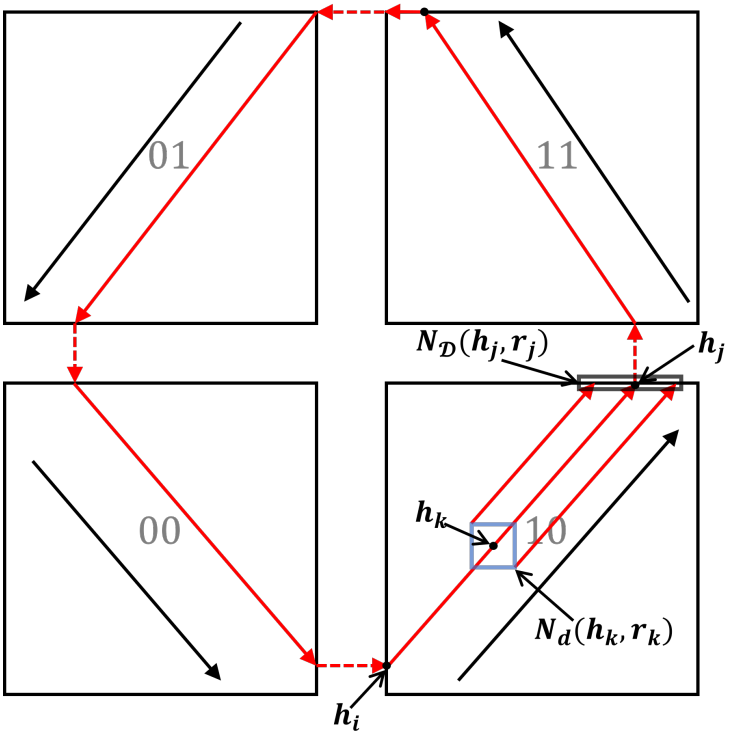


Figure 3.14 – Illustration of Proposition 3. h_i and h_j are two adjacent turning states on a periodic hybrid trajectory. h_k is a hybrid state on the hybrid trajectory from h_i to h_j. Black box represents a neighborhood in the same discrete domain of h_j and blue box represents a neighborhood in the same discrete state of h_k. These neighborhoods correspond to Proposition 3.

By using triangle inequality, we can get a sufficient condition of Equation 3.18, described by inequalities of the sum of absolute values of these δ:

$$\begin{aligned} \mid \delta_1 \mid + \mid \delta_2 \mid &< \frac{\frac{1}{c_s^2}}{\max(\frac{1}{|c_s^1|}, \frac{1}{|c_s^2|})} \\ \mid \delta_1 \mid + \mid \delta_3 \mid &< \frac{\frac{1-\pi_i^3}{c_s^3}}{\max(\frac{1}{|c_s^1|}, \frac{1}{|c_s^3|})} \\ \mid \delta_1 \mid + \mid \delta_4 \mid &< \frac{\frac{1-\pi_i^4}{c_s^4}}{\max(\frac{1}{|c_s^1|}, \frac{1}{|c_s^4|})} \end{aligned} \tag{3.19}$$

These inequalities are called Constraint 2.

By combing Constraint 1 and Constraint 2, same as in the proof of Proposition 1 ,we can get a $N_d(h_i, r_i)$ such that all hybrid trajectories from $N_d(h_i, r_i)$ reach $N_{\mathcal{D}}(h_j, r_j)$. In this example, there is only one sliding boundary (the first dimension) between h_i and h_j. If there are several sliding boundaries, to compute the constraints of δ, we need to ensure that any of these sliding boundaries is reached before any non sliding boundary, so there will be more inequalities in Eq (3.18). But we can always find a sufficient condition of these inequalities which is described by inequalities of the sum of absolute values of these δ.

For any other hybrid state h_k in τ_i (not including h_j), we can use exactly the same method to find such a $N_d(h_k, r_k)$. □

Proposition 4 *Consider a periodic hybrid trajectory τ with the sequence of turning states $T_s(\tau) = (h_1, h_2, ..., h_m)$. $h_{i_0}, h_{i_0+1}, ..., h_{j_0}$ is a sequence in $T_s(\tau)$ such that $\forall i \in \{i_0, i_0+1, ..., j_0-2\}$ the transition from h_i to h_{i+1} is an instant transition and the transition from h_{j_0-1} to h_{j_0} is a non-instant transition (such sequence exists because in this chapter we do not consider periodic hybrid trajectories with only instant transitions). For any neighborhood in the same discrete domain of h_{j_0}: $N_{\mathcal{D}}(h_{j_0}, r_{j_0})$ and for any hybrid state h_i, $i \in \{i_0, i_0+1, ..., j_0-2\}$, if Assumption 1 and Assumption 2 are satisfied, then there exists a neighborhood in the same discrete state of h_i: $N_d(h_i, r_i)$, such that all hybrid trajectories from $N_d(h_i, r_i)$ reach $N_{\mathcal{D}}(h_{j_0}, r_{j_0})$.*

Proof: Without loss of generality, we consider four turning states h_1, h_2, h_3, h_4, where $h_1 = (\pi_1 = (1, 1, 1, \pi_1^4), d_{s_1} = (a, b, c, d))$, $h_2 = (\pi_2 = (0, 1, 1, \pi_1^4), d_{s_2} = (a+1, b, c, d))$,

$h_3 = (\pi_3 = (0, 0, 1, \pi_1^4), d_{s_3} = (a+1, b+1, c, d))$, $h_4 = (\pi_4 = (\pi_4^1, \pi_4^2, \pi_4^3, 1), d_{s_3} = (a+1, b+1, c, d))$. We can see that there is an instant transition from h_1 to h_2 which crosses the upper boundary in the first dimension, an instant transition from h_2 to h_3 which crosses the upper boundary in the second dimension and a non-instant transition from h_3 to h_4. The celerity vector of d_{s_1} is $c_{s_1} = (c_s^1, c_s^2, c_s^3, c_s^4)$ and we suppose that all values in c_{s_1} are positive. The celerity vector of d_{s_2} is $c_{s_1} = (c_s^{1\prime}, c_s^{2\prime}, c_s^3, c_s^4)$ and we also suppose that $c_s^{1\prime}$ and $c_s^{2\prime}$ are positive. Consider the neighborhood in the same discrete domain of h_4: $N_{\mathcal{D}}(h_4, r_4)$. Any hybrid state in the neighborhood in the same discrete state of h_1 can be represented by $h_1' = (\pi_1' = (1+\delta_1, 1+\delta_2, 1+\delta_3, \pi_1^4+\delta_4), d_{s_1} = (a, b, c, d))$.

We suppose that in d_{s_1}, the upper boundaries in the second and third dimension are attractive boundaries and the upper boundary in the fourth dimension is an output boundary (there is only one reached output boundary because of Assumption 2). After the upper boundary e_1 in the first dimension in d_{s_1} is crossed, in d_{s_2} the upper boundary e_2 in the second dimension becomes an output boundary (e_1 is related to a threshold of first gene that influences the second gene), so e_2 is crossed by the hybrid trajectory.

We want that the hybrid trajectory from h_1' will firstly cross the upper boundary in the first dimension and then cross the upper boundary in the second dimension. To ensure this, we only need to ensure that e_1 and e_2 are reached before the upper boundary e_4 in the fourth dimension is reached. We do not need to ensure that e_1 is reached before e_2 because e_2 is an attractive boundary. We need to satisfy the following constraints:

$$\begin{aligned} -\frac{\delta_1}{c_s^1} &< \frac{1-\delta_4-\pi_i^4}{c_s^4} \\ -\frac{\delta_2}{\min(c_s^2, c_s^{2\prime})} &< \frac{1-\delta_4-\pi_i^4}{c_s^4} \end{aligned} \tag{3.20}$$

A sufficient condition, for these constraints, represented by inequalities of the sum of absolute values of δ, to be satisfied, is:

$$\begin{aligned} \mid \delta_1 \mid + \mid \delta_4 \mid &< \frac{\frac{1-\pi_i^4}{c_s^4}}{\max(\frac{1}{|c_s^1|}, \frac{1}{|c_s^4|})} \\ \mid \delta_2 \mid + \mid \delta_4 \mid &< \frac{\frac{1-\pi_i^4}{c_s^4}}{\max(\frac{1}{|\min(c_s^2, c_s^{2\prime})|}, \frac{1}{|c_s^4|})} \end{aligned} \tag{3.21}$$

This condition is noted Constraint 1.

After the hybrid trajectory from h'_1 crosses e_2, the hybrid state can be represented by $h'_3 = (\pi'_3 = (0 + \Delta_1, 0, 1 + \Delta_3, \pi_1^4 + \Delta_4), d_{s3} = (a + 1, b + 1, c, d))$ where Δ_1, Δ_3 and Δ_4 are linear combinations of δ_1, δ_2, δ_3 and δ_4. To ensure that the hybrid trajectory from h'_3 reaches $N_{\mathcal{D}}(h_4, r_4)$, using a similar method as in the proof of Proposition 3, we can get constraints represented by inequalities of the sum of absolute values of these δ (like Eq (3.16)). By combining these constraints with Constraint 1, we can get $N_d(h_1, r_1)$ such that all hybrid trajectories from $N_d(h_1, r_1)$ reach $N_{\mathcal{D}}(h_4, r_4)$.

The above demonstration only considers one specific example, but this method can be used to find such $N_d(h_i, r_i)$ in any case if Assumption 1 and Assumption 2 are satisfied. In this example, we consider a sequence of two instant transitions. If there is a sequence of N instant transitions, we only need to ensure that any boundary, that is crossed in the sequence, is reached before any other boundary is reached, so we have more inequalities in Eq (3.20). □

Now based on the above propositions, we have the following theorem.

Theorem 2 *The neighborhood of a periodic hybrid trajectory is continuous if Assumption 1 and Assumption 2 are satisfied.*

Proof: Consider any neighborhood in the same discrete domain of a turning state h_0 of τ: $N_{\mathcal{D}}(h_0, r_0)$ and consider any h_1 on τ.

If h_1 is not a turning state of τ, then we can find two adjacent turning states h_i and h_j such that there is a non-instant transition between h_i and h_j (a straight hybrid trajectory) and h_1 belongs to this straight hybrid trajectory. According to Proposition 2, we can find a neighborhood in the same discrete domain of h_j: $N_{\mathcal{D}}(h_j, r_j)$, such that all hybrid trajectories from $N_{\mathcal{D}}(h_j, r_j)$ will finally reach $N_{\mathcal{D}}(h_0, r_0)$. Then according to Proposition 3, we can find a neighborhood in the same discrete state of h_1: $N_d(h_1, r_1)$, such that all hybrid trajectories from $N_d(h_1, r_1)$ will reach $N_{\mathcal{D}}(h_j, r_j)$. So all hybrid trajectories from $N_d(h_1, r_1)$ will firstly reach $N_{\mathcal{D}}(h_j, r_j)$ and finally reach $N_{\mathcal{D}}(h_0, r_0)$.

If h_1 is a turning state of τ and the transition following h_1 is a non-instant transition, that means that there is a straight hybrid trajectory from h_1 that reaches the next turning state h_2. Then, same as the previous case, we can find a neighborhood in the same discrete state of h_1: $N_d(h_1, r_1)$, such that all hybrid trajectories from $N_d(h_1, r_1)$ will finally reach $N_{\mathcal{D}}(h_0, r_0)$.

If h_1 is a turning state of $\mathcal{C}_\tau$ and the transition following h_1 is an instant transition, that means that the hybrid trajectory from h_1 firstly crosses a boundary and reaches the

next turning state h_2. We note h_j as the first turning state after h_2 such that the transition before h_j is a non-instant transition. According to Proposition 2, we can find a neighborhood in the same discrete domain of h_j: $N_{\mathcal{D}}(h_j, r_j)$, such that all hybrid trajectories from $N_{\mathcal{D}}(h_j, r_j)$ will finally reach $N_{\mathcal{D}}(h_0, r_0)$. Then according to Proposition 4, we can find a neighborhood in the same discrete state of h_1: $N_d(h_1, r_1)$, such that all hybrid trajectories from $N_d(h_1, r_1)$ will reach $N_{\mathcal{D}}(h_j, r_j)$. So all hybrid trajectories from $N_d(h_1, r_1)$ will firstly reach $N_{\mathcal{D}}(h_j, r_j)$ and finally reach $N_{\mathcal{D}}(h_0, r_0)$.

Now we have proved that for any h_1 in τ, we can find a neighborhood in the same discrete state of h_1: $N_d(h_1, r_1)$, such that all hybrid trajectories from $N_d(h_1, r_1)$ will finally reach $N_{\mathcal{D}}(h_0, r_0)$. So the neighborhood of τ is continuous.

□

3.3.2 Eigenanalysis

In this subsection, we present our method to analyze the stability of limit cycles. Consider a periodic hybrid trajectory τ inside the closed discrete trajectory $\mathcal{T} = (\mathcal{D}_1, \mathcal{D}_2, ...\mathcal{D}_m, \mathcal{D}_1)$. Suppose that τ reaches $\mathcal{D}_1$ at $h = (\pi, d_{s_1})$, thus we have:

$$\pi = s^{-1}(M_{\mathcal{T}} s(\pi)) \tag{3.22}$$

For π, there might be some dimensions in which the values are 0 or 1 because in these dimensions the upper or lower boundaries are reached. If we only consider the dimensions in which the boundaries are not reached, Eq (3.22) becomes:

$$x = Ax + b \tag{3.23}$$

where x, called the *reduction vector* of h (denoted by $x = r(h)$), is a short version of π which only contains the dimensions in which the boundaries are not reached. The matrix A is called the *reduction matrix* of $\mathcal{T}$ and vector b is called the *constant vector* of $\mathcal{T}$. Based on the concept of reduction vector, the compatible zone of a sequence of discrete domains $\mathcal{T} = (\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m)$ can be described by $\mathcal{S} = \{(\pi, d_{s_0}) \in \mathcal{D}_0 \mid r(\pi) \in \mathcal{S}_r\}$ where $\mathcal{S}_r$, called the *reduction compatible zone*, is a set of reduction vectors of hybrid states in $\mathcal{D}_0$.

If the length of x is not 0, meaning that $\mathcal{D}_1$ does not contain only a singular hybrid state (a counter example is the discrete domain 1^+1^+ in Fig 3.5 which contains only a singular hybrid state), then the stability analysis method of the limit cycles is based on

Theorem 3.

Theorem 3 *Consider a periodic hybrid trajectory τ inside the closed discrete trajectory $\mathcal{T} = (\mathcal{D}_1, \mathcal{D}_2, ...\mathcal{D}_m, \mathcal{D}_1)$, and $\lambda_1, \lambda_2, ..., \lambda_p$ the eigenvalues of the reduction matrix A of $\mathcal{T}$.*

- *If $\max_{i\in\{1,2,...,p\}}|\lambda_i| < 1$ then τ is a stable limit cycle.*
- *If $\max_{i\in\{1,2,...,p\}}|\lambda_i| > 1$ then τ is an unstable limit cycle.*
- *If $\max_{i\in\{1,2,...,p\}}|\lambda_i| = 1$ and $\exists i_0 \in \{1, 2, ..., p\}, |\lambda_{i_0}| < 1$, then τ is an unstable limit cycle.*
- *If $\forall i \in \{1, 2, ..., p\}, |\lambda_i| = 1$, then τ is not a limit cycle.*

Proof: Consider a periodic hybrid trajectory τ that exists inside a closed discrete trajectory $\mathcal{T}$. The intersection of τ with the chosen Poincaré section e is $h_0 = (\pi_0, d_{s_0})$. The Poincaré map is noted as $x^{k+1} = Ax^k + b$, where x is the reduction vector of the hybrid states in the compatible zone of $\mathcal{T}$ ($x_0 = r(h_0)$). The stability of the fixed point(s) of the system $x^{k+1} = Ax^k + b$ depends on the eigenvalues of A.

If the absolute values of all eigenvalues of A are less than 1, then x_0 is asymptotically stable for the system $x^{k+1} = Ax^k + b$. And since the neighborhood of τ is continuous, we can find a small neighborhood in the same discrete domain of h_0: $N_{\mathcal{D}}(h_0, r_0)$, such that any hybrid trajectory from $N_{\mathcal{D}}(h_0, r_0)$ stays inside the neighborhood of τ and converges asymptotically to or reaches τ. Also, based on the fact that the neighborhood of τ is continuous, for any hybrid state h' on τ, we can find a neighborhood in the same discrete state of h': $N_d(h', r)$, such that any hybrid trajectory from $N_d(h', r)$ reaches $N_{\mathcal{D}}(h_0, r_0)$. Thus, for any hybrid trajectory τ' from $N_d(h', r)$, we have: $\lim_{t\to\infty} Dis_{min}(\tau'(t), \tau) = 0$, which proves that τ is a stable limit cycle.

If the maximum absolute value of all eigenvalues of A is greater than 1, then x_0 is unstable for system $x^{k+1} = Ax^k + b$, so we can always find a hybrid trajectory from any small neighborhood in the same discrete domain of h_0 that does not converge to or reach τ. Therefore, τ is unstable.

If the maximum absolute value of all eigenvalues of A is 1 and there is at least one eigenvalue that differs from 1, then in any small neighborhood in the same discrete domain of h_0, we can find hybrid trajectories that do not converge to or reach τ (which might converge to or reach another limit cycle), so τ is unstable.

If the absolute values of all eigenvalues are 1, then all hybrid trajectories from any small neighborhood in the same discrete domain of h_0 are periodic hybrid trajectories (like Fig 3.4), so τ is not a limit cycle.

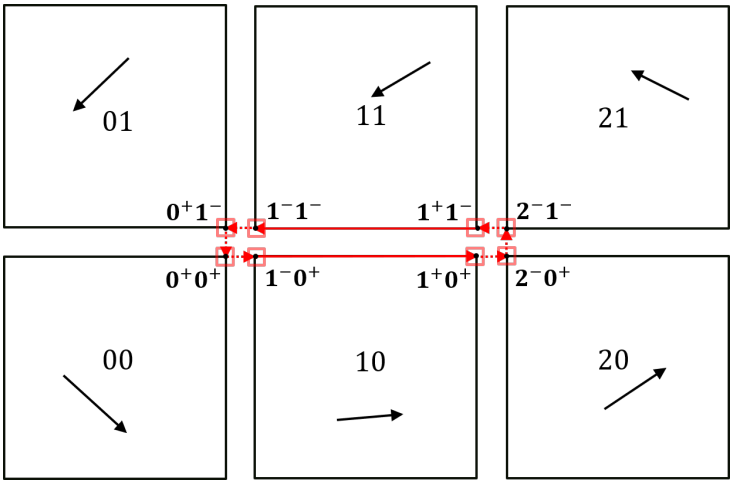


Figure 3.15 – Example of a limit cycle inside a sequence of discrete domains where each discrete domain contains only a singular hybrid state.

□

In case that the length of x is 0, in other words $\mathcal{D}_1$ contains only a singular hybrid state, if there exists $i \in \{1, 2, ..., m\}$ such that $\mathcal{D}_i$ does not contain only a singular hybrid state, then $\mathcal{D}_i$ can be chosen as the new $\mathcal{D}_1$ to apply Theorem 3. If any discrete domain in this sequence contains only a singular hybrid state (see the example in Fig 3.15), then we have directly that τ is a stable limit cycle because the neighborhood of τ is continuous.

3.4 Applications

In this section, we apply the above limit cycle analysis method on three HGRNs of negative feedback loop in 3 dimensions and one HGRN of cell cycle in 5 dimensions. The negative feedback loop in 3 dimensions can be used to describe real biological oscillators, for example the p53 system [41]. The signs of the celerities in these three HGRNs are determined by the influence graph (positive for an activation and negative for an inhibition) and their absolute values of celerities are randomly selected. The parameters of the HGRN in 5 dimensions are generated randomly respecting the constraints in Table 3 of [10]. The influence graphs of both systems can be found in Fig 3.16. Details about the implementation can be found at `https://doi.org/10.5281/zenodo.6524936`.

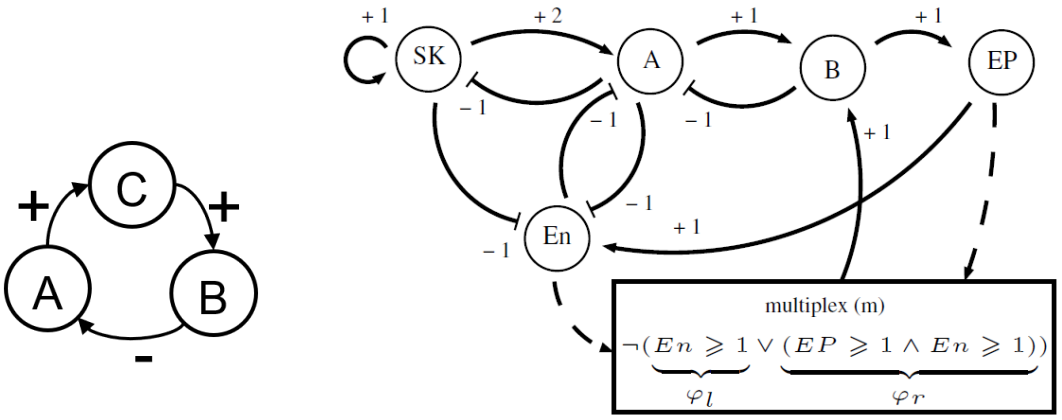


Figure 3.16 – Left: Influence graph of a negative feedback loop with 3 genes, used to build three models given in Table 3.1. Right: Influence graph of a cell cycle model with 5 genes from [10]; the multiplex (m) expresses constrains on the joint activation of En and Ep on B.

Table 3.1 – Parameters of the three HGRNs of negative feedback loop in 3 dimensions. Left: First model. Middle: Second model. Right: Third model.

A	B	C	C_A	C_B	C_C
0	0	0	1	-0.6	-0.7
0	0	1	1	0.7	-0.9
0	1	0	-0.8	-0.8	-0.7
0	1	1	-0.8	0.6	-0.9
1	0	0	0.7	-0.6	0.6
1	0	1	0.7	0.7	0.5
1	1	0	-0.9	-0.8	0.6
1	1	1	-0.9	0.6	0.5

A	B	C	C_A	C_B	C_C
0	0	0	3	-0.6	-0.7
0	0	1	3	0.7	-2.9
0	1	0	-2.8	-0.8	-0.7
0	1	1	-2.8	0.6	-2.9
1	0	0	2.7	-0.6	2.6
1	0	1	2.7	0.7	0.5
1	1	0	-2.9	-0.8	2.6
1	1	1	-2.9	0.6	0.5

A	B	C	C_A	C_B	C_C
0	0	0	3	-0.6	-0.7
0	0	1	3	0.7	-2.9
0	1	0	-0.8	-0.8	-0.7
0	1	1	-0.8	0.6	-2.9
1	0	0	0.7	-0.6	2.6
1	0	1	0.7	0.7	0.5
1	1	0	-2.9	-0.8	2.6
1	1	1	-2.9	0.6	0.5

3.4.1 3-dimensional Models of Negative Feedback Loop

The parameters of these three HGRNs of negative feedback loop in 3 dimensions are shown in Table 3.1. The signs of the celerities in these three models are the same so they have the same graph of discrete states. There is only one cycle of discrete states in each of these systems, which is:

$111 \to 011 \to 010 \to 000 \to 100 \to 101 \to 111$

Therefore, for these three models, we choose the input boundary e of 000 in the cycle as the Poincaré section. Simulations depicting the convergence to the stable cycle or to the fixed point (see below) in these three HGRNs are shown in Fig 3.17.

In the first HGRN, by using our limit cycle analysis method, we find one sta-

ble limit cycle and one periodic hybrid trajectory which only contains instant transitions, which is considered here as a fixed point of this system. Regarding the stable limit cycle, the fixed point of this limit cycle in discrete domain 0^-0^+0 is $((0,1,0.125),(0,0,0))$, the transition matrix is $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0.125 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, the compatible zone is $\{(\pi,(0,0,0)) \mid \pi^1 = 0, \pi^2 = 1, \pi^3 \in]0, 0.7[\}$ and the reduction matrix is $\begin{bmatrix} 0 \end{bmatrix}$, therefore hybrid trajectories from the neighborhood of this limit cycle will reach this limit cycle very quickly (less than one turn if the neighborhood is small enough).

In this HGRN we can also prove that all hybrid trajectories will reach this limit cycle except hybrid trajectories which can reach the fixed point of the system. All discrete trajectories which begin from the Poincaré section and return to the Poincaré section in this HGRN are shown in Fig 3.18 A. Since in this HGRN there is only one cycle of discrete states which is also a global attractor, any hybrid trajectory from the Poincaré section must return to the Poincaré section and it must begin from the compatible zone or the boundary of the compatible zone of one of the discrete trajectories in Fig 3.18 A. We see that all discrete trajectories which are not closed will finally reach closed discrete trajectories (31, 32, 33, 9, 10). Discrete trajectories 31, 32 and 33 have the same transition matrix and their reduction matrix is $\begin{bmatrix} 0 \end{bmatrix}$ so any hybrid trajectory from 0^-0^+0 will reach the limit cycle. For the discrete trajectory 10, the two eigenvalues of the reduction matrix are 7.0306 and 0.0368, so hybrid trajectories inside discrete trajectory 10 will finally leave the compatible zone and reach 0^-0^+0. An example of such hybrid trajectory is illustrated in Fig 3.19, this hybrid trajectory starts at the hybrid state h_1 on the Poincaré section and returns to the Poincaré section for the first time at hybrid state h_2. The compatible zone of the discrete trajectory 10 is illustrated by the red triangle (not including the boundaries) and v_1 and v_2 represent the two eigenvectors. Since the absolute value of the first eigenvalue is bigger than 1 and the absolute value of the second eigenvalue is smaller than 1, h_2 is farther from the up-right corner (which is a fixed point on the Poincaré section) in the direction of v_1 and is closer to the up-right corner in the direction of v_2, so this hybrid trajectory leaves the compatible zone of the closed discrete trajectory 10 and reaches the compatible zone of the closed discrete trajectory 11. From here, we can see that any hybrid trajectories from the Poincaré section will reach the limit cycle except the hybrid trajectory inside discrete trajectory 9 which is related to a fixed point. As

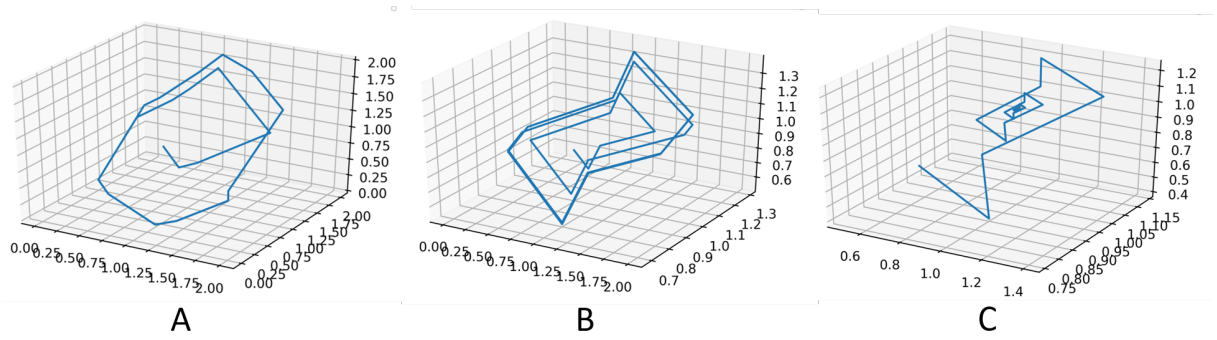


Figure 3.17 – Illustration of stable limit cycles and stable fixed point in HGRNs in 3 dimensions. A: Stable limit cycle in the first HGRN. B: Stable limit cycle in the second HGRN. C: Stable fixed point in the third HGRN.

any hybrid trajectory in this system will finally reach this Poincaré section, all hybrid trajectories will reach this limit cycle except hybrid trajectories which can reach the fixed point.

For the second HGRN, by using our method, we can also find one stable limit cycle and one fixed point. Unlike the first HGRN, hybrid trajectories from the neighborhood of the limit cycle converge asymptotically to the limit cycle: The limit cycle is inside the discrete trajectory which begins from 0^-0^+0 and the reduction matrix of the limit cycle is $\left[0.0298\right]$. We can also prove that all hybrid trajectories will converge to this limit cycle except hybrid trajectories which can reach the fixed point, by using the same method as for the first HGRN.

Contrary to the first and the second HGRN, we cannot find a limit cycle in the third HGRN but only a fixed point which is related to the discrete trajectory 2 in Fig 3.18 C. For the discrete trajectories 3 and 4, the fixed point of the Poincaré map of discrete trajectories 4 is strictly inside the compatible zone of discrete trajectory 3, the fixed point of the Poincaré map of the discrete trajectory 3 is the fixed point of this HGRN and the absolute values of all eigenvalues of the reduction matrix of discrete trajectories 3 and 4 are strictly inferior than 1 (the two eigenvalues of the reduction matrix of discrete trajectory 3 are 0.3091 and 0.0062, and the two eigenvalues of the reduction matrix of the discrete trajectory 4 are 0 and 0.1269), so we can prove that all hybrid trajectories in this system will converge to the fixed point. An illustration of this fixed point is given in Fig 3.17 C.

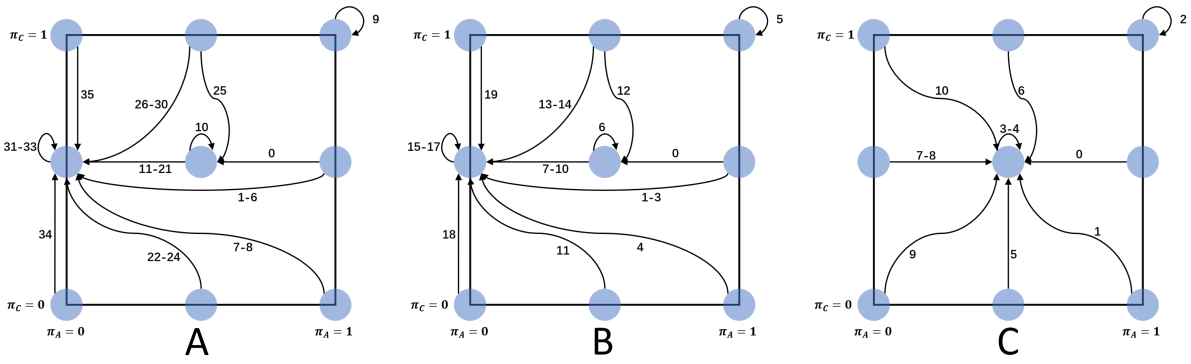


Figure 3.18 – Abstracted representations of the chosen Poincaré sections in the HGRNs in 3 dimensions, illustrating all possible discrete trajectories which start from and return to this Poincaré section. The blue dots represent the discrete domains and each arrow depicts one or several different discrete trajectories (each following a unique sequence of discrete domains). A: First HGRN. B: Second HGRN. C: Third HGRN.

3.4.2 5-dimensional Model of Cell Cycle

For the HGRN in 5 dimensions, the transition graph of discrete states is more complex (Fig 3.20). By using a depth first algorithm, we find that there are 1104 cycles of discrete states in which 930 cycles contain the discrete transition 01010 $\rightarrow$ 01011, 94 cycles contain 00011 $\rightarrow$ 01011 and all the rest contain 00110 $\rightarrow$ 00010. So for this system, we use 3 Poincaré sections which are the input boundary of 01011 between 01010 and 01011, the input boundary of 01011 between 00011 and 01011 and the input boundary of 00010 between 00110 and 00010. Any limit cycle of this system must cross one of these Poincaré sections. By using the method above, we find 184 closed discrete trajectories which cross the first Poincaré section, among these 184 closed discrete trajectories, there is only one stable limit cycle; we find 50 closed discrete trajectories which cross the second Poincaré section, these 50 closed discrete trajectories do not contain any limit cycle; and we find 111 closed discrete trajectories which cross the third Poincaré section, among these 111 closed discrete trajectories, there is one stable limit cycle and one unstable limit cycle. In fact, these two stable limit cycles are the same which crosses both the first and the third Poincaré sections. This stable limit cycle is the same one studied in [10] to calculate the constraints of parameters. The simulations of both cycles are shown in Fig 3.21 A and B. For now we have not identified any biological behavior related to this unstable limit cycle yet.

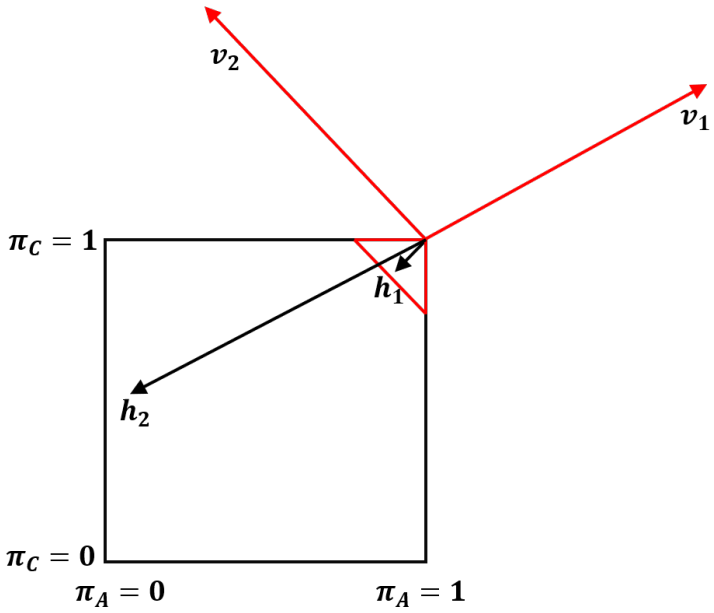


Figure 3.19 – Illustration of a hybrid trajectory inside the closed discrete trajectory 10 in Fig 3.18 A on the Poincaré section. Red triangle represents the compatible zone (not including boundaries). v_1 and v_2 represent the two eigenvectors. Hybrid trajectory from hybrid state h_1 on the compatible zone returns to the Poincaré section at hybrid state h_2.

For the stable limit cycle of the cell cycle model, the fixed point of this limit cycle in the discrete domain $01^+01^+1^-$ is $((0.3714, 1, 0.8581, 1, 0), (0, 1, 0, 1, 1))$, and the reduction matrix is $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$.

For the unstable limit cycle of cell cycle model, the fixed point of this limit cycle in the discrete domain 000^+10^- is $((0.6375, 0.2552, 1, 0.3472, 0), (0, 0, 0, 1, 0))$, and the reduction matrix A is $\begin{bmatrix} 4.95359512 \cdot 10^3 & 0 & 1.37489884 \cdot 10^{-13} \\ -5.25996267 \cdot 10^2 & 0 & -1.45993292 \cdot 10^{-14} \\ -7.15619779 \cdot 10^2 & 0 & -1.98624389 \cdot 10^{-14} \end{bmatrix}$. The eigenvalues of A are 0, $4.95359512 \cdot 10^3$ and $3.15544362 \cdot 10^{-30}$, making it unstable.

The current implementation in Python reaches its limits with respect to execution time when the size of the system increases: Finding the limit cycles above takes less than one minute for the HGRNs in 3 dimensions, and 8 hours for the HGRN in 5 dimensions [1].

1. Computations were performed on a standard laptop computer, with an Intel Core I7-8550U 1.80GHz processor and 16.0GB RAM.

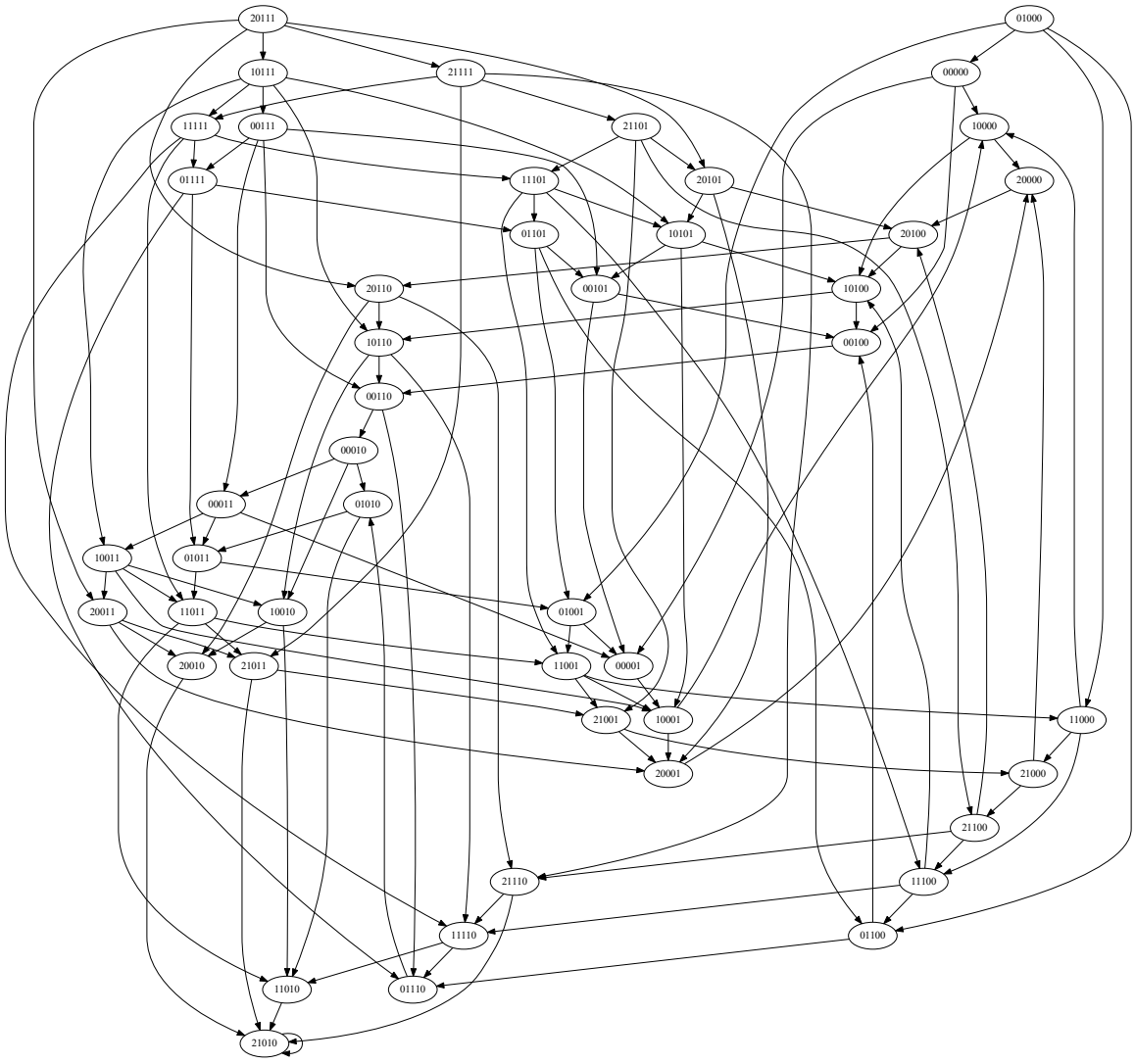


Figure 3.20 – Transition graph of discrete states of the HGRN of cell cycle in 5 dimensions.

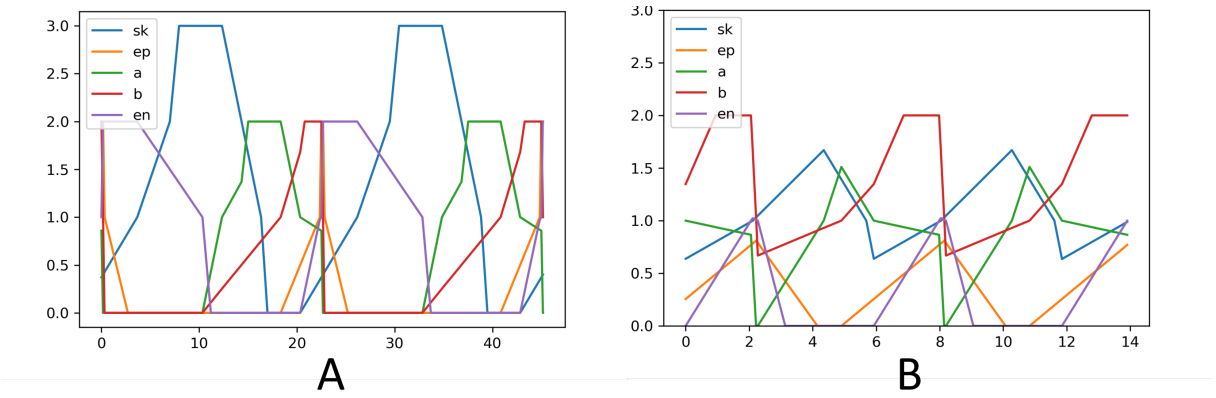


Figure 3.21 – Simulation of the two limit cycles found in the HGRN of 5 dimensions. A: Stable limit cycle. B: Unstable limit cycle.

3.5 Summary

This chapter presented a method to find all limit cycles of HGRNs with some minor restrictions, mainly to remove non-deterministic behaviors and complex loops, and to analyze their stability. This method is the first one to find and analyze limit cycles of HGRNs in N dimensions. We showed the merits of this method on randomly generated HGRNs of a negative feedback loop with 3 components and a HGRN of the cell cycle with 5 components taken from the literature.

As stated above, a first limitation of this method is that we do not handle complex periodic hybrid trajectories which are periodic hybrid trajectories crossing a discrete state for more than once in one period. Such periodic hybrid trajectories are not considered because they are hard to enumerate.

Another limitation in the application of this method is that we first need to construct a HGRN of a specific gene regulatory network; however, the observation of real biological systems is limited and it is not always possible to determine all parameters. In some cases, some parameters can only be described by constraints, or remain unknown. Thus, considering extensions of this method that are parameterized or that take into account a set of constraints on parameters is of interest. Another work about analysis of oscillations in a parameterized HGRN of a specific network is presented in chapter 5.

In this chapter, we focused on the existence of limit cycles and their stability, which is one way to describe the dynamical properties of a system. They are of particular interest because they represent the long-term behaviors of the system. In the next chapter, we

will continue to use the new concepts introduced in this chapter and present our analysis method of reachability, which is an another kind of dynamical properties.

Chapter 4

Reachability Analysis

In this chapter, a new reachability method on HGRNs is presented. The reachability problem concerned in this chapter is, given a singular hybrid state and a region (a set of hybrid states), to determine whether the hybrid trajectory from this singular hybrid state will reach this region. This problem is undecidable for general hybrid automata, and is known to be decidable only for a restricted class of hybrid automata, but this restricted class does not include HGRNs. So, a priori, this reachability problem in HGRNs is not decidable; however, we find that it is decidable in some cases. Based on this fact, the main idea of this work is that if the decidable cases can be determined automatically, then the reachability problem can be solved partially. The method of this chapter relies on the concepts introduced in the previous chapter, e.g., transition matrix, discrete domain, compatible zone, etc. The two major contributions are the following: firstly, we classify hybrid trajectories into different classes and provide theoretical results about decidability; then based on these theoretical results, we propose a reachability analysis algorithm which always stops in finite time and answers partially the reachability problem, meaning that it gives the correct answer if it is not inconclusive. Finally, by applying our implementation of this method on a 5-dimensional HGRN of cell cycle, we show a potential application of this method to estimate the basin of attraction. This work has been presented at RP 2023 as regular paper [14].

4.1 Problem Statement

One important question concerning the analysis of dynamical system is reachability: Whether a certain state (or set of states) is reachable from an initial state (or set of states), which is widely used to describe the dynamical properties of gene regulatory networks [42, 43].

Reachability analysis methods have been studied on different formalisms, mainly on discrete systems [44–46] and hybrid systems [23, 42, 47–51]. In this chapter, we study a

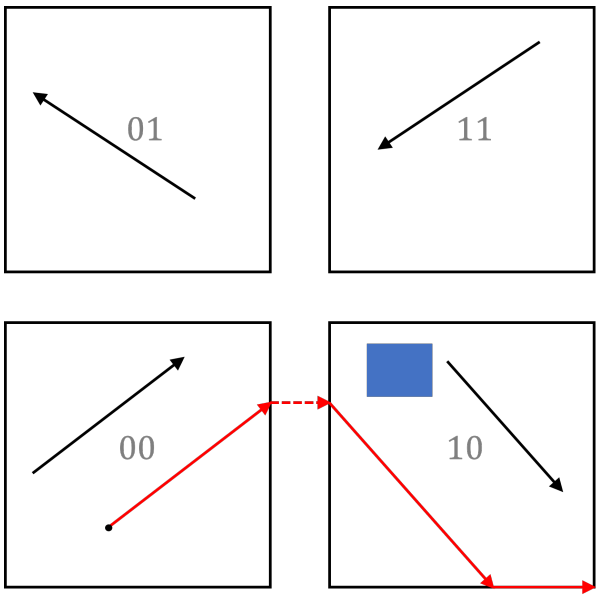


Figure 4.1 – Illustration of Problem 1 and hybrid trajectory halting in finite time. Red arrows represent the hybrid trajectory from h_{ini}. The blue rectangle represents R_{target}.

reachability analysis method of HGRNs.

There are generally two kinds of reachability problems in hybrid systems: The first one is, given two states (or sets of states) A and B, to determine whether trajectories (or a trajectory) from A reach(es) B [23, 48, 51, 52]; the second one is to compute all states reachable from certain initial states [49, 50, 53, 54]. The reachability problem concerned in this chapter belongs to the first case, which is to determine whether the hybrid trajectory from certain hybrid state can reach a certain region (a set of hybrid states) with the restriction that we only consider single hybrid trajectories, that is, starting from a single hybrid state and avoiding any non-determinism. The reachability problem of this work is defined formally as follows, where we assume that the system has N genes (N dimensions).

Problem 1 (Reachability) *Consider a hybrid state* $h_{ini} = (\pi_{ini}, d_{s_{ini}})$ *and a region* $R_{target} = \left\{(\pi, d_{s_{target}}) \mid \pi^i \in [a_i, b_i], i \in \{1, 2, ..., N\}\right\}$, *where* $\forall i \in \{1, 2, ..., N\}, a_i, b_i \in \mathbb{R}$ *and* $0 \leq a_i \leq b_i \leq 1$. *Does the hybrid trajectory* τ *from* h_{ini} *enter the region* R_{target}*? In other words, does there exist* t_0 *such that* $\tau(t_0) \in R_{target}$*?*

Problem 1 is illustrated in the examples of Fig 4.1, where the initial hybrid state of the hybrid trajectory (the hybrid trajectory is illustrated by red arrows) is h_{ini} and the blue rectangle represents R_{target}.

The following assumptions are made in this chapter.

Assumption 3 *For any sequence of discrete domains $\mathcal{T}$ of which the compatible zone is not empty, we assume that all eigenvalues of the reduction matrix of $\mathcal{T}$ are real.*

For now, we have not found a reduction matrix with complex eigenvalues. For the definition of reduction matrix, see Section 3.3.2.

Assumption 4 *The hybrid trajectory from h_{ini} has no non-deterministic behavior.*

Generally, hybrid trajectories with non-deterministic behaviors exist, but among state-of-the-art HGRNs of gene regulatory networks, the probability of a randomly chosen initial hybrid state that leads to non-deterministic behaviors is almost 0. Therefore, we ignore this kind of hybrid trajectory in this chapter. Actually, the method of this work could also be adapted for non-deterministic hybrid trajectories (each time when a non-deterministic hybrid state is reached, the current hybrid trajectory splits into two or several hybrid trajectories, and the same method is applied on each of these new hybrid trajectories).

Assumption 5 *Any non-instant transition on a limit cycle does not reach more than one new boundary at the same time.*

This assumption is very similar to Assumption 1 in Chapter 3. In fact, Assumption 5 is less restrictive than Assumption 1, because there exist periodic hybrid trajectories that are not limit cycles and have non-instant transition reaching more than one new boundary at the same time; HGRNs with such periodic hybrid trajectories are ignored in Chapter 3 but are not ignored in this chapter.

In this chapter, we mainly focus on the decidability problem, that is, whether we can find an algorithm to determine the reachability problem such that this algorithm always stops in finite time and gives a correct answer.

The decidability problem among some hybrid systems that are close to HGRNs has been investigated in the literature. It has been proved that, for PCD systems (Piecewise-Constant Derivative systems), it is decidable in 2 dimensions [22] but it is undecidable in 3 dimensions [23]. For general hybrid automata, there exists a restricted class called initialized rectangular automata which is decidable in any dimension [47], but this class does not include HGRNs.

Up to now, there is no theoretical result of the decidability of the reachability problem on HGRNs. A priori, we can expect that it is not decidable because of the existence of chaos. However, if we can show that it is decidable in certain cases, for example, when

the hybrid trajectory considered in a reachability problem converges asymptotically to a n-dimensional limit cycle, and if these cases can be identified automatically, then the reachability problem can be answered partially. Based on this idea, we classified the hybrid trajectories of HGRNs into different classes which are introduced in the next section.

4.2 Different Classes of Hybrid Trajectories

We classify hybrid trajectories of HGRNs into three classes: Hybrid trajectories halting in finite time, hybrid trajectories attracted by cycles of discrete domains and chaotic hybrid trajectories, which are introduced as follows.

4.2.1 Hybrid Trajectories Halting in Finite Time

A hybrid trajectory τ is a *hybrid trajectory halting in finite time* if $\exists t_0$ such that the time derivative of $\tau(t_0)$ is 0 in any dimension; in other words, $\tau(t_0)$ is a fixed point. Therefore, the discrete domains reached by a hybrid trajectory halting in finite time form a finite sequence like $(\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m)$. The hybrid trajectory in Fig 4.1 is a hybrid trajectory halting in finite time.

Theorem 4 *Problem 1 is decidable if the hybrid trajectory from h_{ini} is a hybrid trajectory halting in finite time.*

Proof: In this case, the hybrid trajectory is a composition of a finite number of n-dimensional "straight lines"; to verify if this hybrid trajectory reaches R_{target}, we only need to verify if any of these "straight lines" crosses R_{target}, which can be verified in finite time. □

4.2.2 Hybrid Trajectories Attracted by Cycles of Discrete Domains

A hybrid trajectory τ is a *hybrid trajectory attracted by a cycle of discrete domains* if $\exists t_0$ such that after t_0, τ always stays inside a cycle of discrete domains $\mathcal{C}_\mathcal{T} = (\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_p, \mathcal{D}_0)$, meaning that τ crosses this cycle for an infinite number of times without leaving it. Intuitively, if a hybrid trajectory τ is attracted by a cycle of discrete domains, then τ converges to or reaches a periodic hybrid trajectory. We

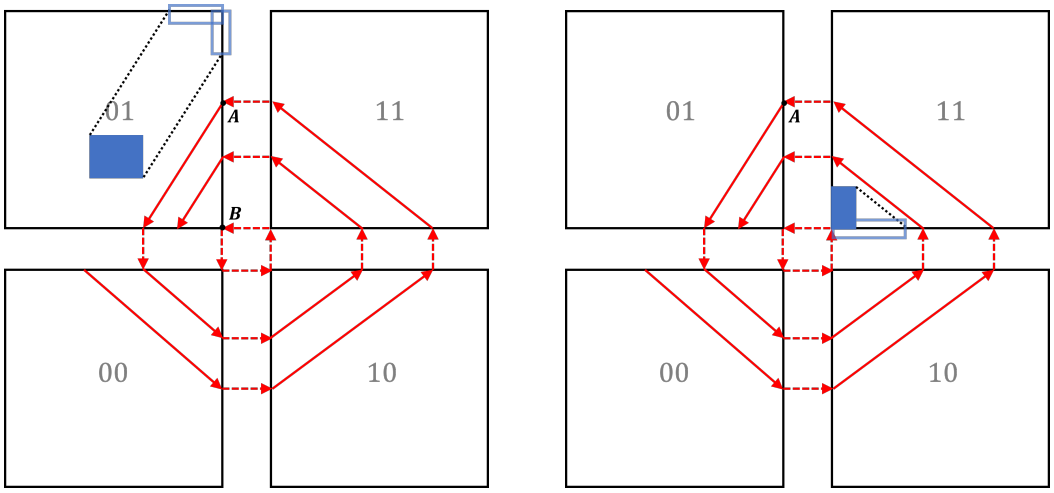


Figure 4.2 – Illustration of hybrid trajectories attracted by cycles of discrete domains and predecessor in the same discrete state. Blues rectangles represent R_{target} of Problem 1 and blue boxes represent their predecessors in the same discrete state.

say that this periodic hybrid trajectory attracts τ. So the sequence of discrete domains reached by a hybrid trajectory attracted by a cycle of discrete domains is of the form: $(\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_k, (\mathcal{D}_{k+1}, \mathcal{D}_{k+2}, ..., \mathcal{D}_{k+p})_n)$, where the notation $(\mathcal{D}_{k+1}, \mathcal{D}_{k+2}, ..., \mathcal{D}_{k+p})_n$ means that the sequence $(\mathcal{D}_{k+1}, \mathcal{D}_{k+2}, ..., \mathcal{D}_{k+p})$ is repeated an infinite number of times.

In Fig 4.2, both hybrid trajectories are attracted by a cycle of discrete domains: Indeed, these hybrid trajectories converge to the limit cycle in the center of the figure (which only has instant transitions).

To prove the decidability of hybrid trajectories attracted by cycles of discrete domains, we introduce the notion of predecessor in the same discrete state.

Definition 13 (Predecessor in the same discrete state) *For any set of hybrid states in the same discrete state defined by* $R = \{(\pi, d_s) \mid \pi \in E\}$ *where* $E \subseteq [0,1]^N$ *is a closed set, the predecessor of* R *in the same discrete state, noted by* $Pre_{d_s}(R)$*, is the union of sets of hybrid states:* $Pre_{d_s}(R) = \bigcup_{i \in \{1,2,...,q\}} Z_i$*, such that:*

- $\forall i \in \{1, 2, ..., q\}$, $\exists \mathcal{D}_i$ *(a discrete domain) on an input boundary of* d_s*, such that* $Z_i \subseteq \mathcal{D}_i$.
- $\forall i, j \in \{1, 2, ..., q\}, i \neq j$, $\mathcal{D}_i$ *and* $\mathcal{D}_j$ *are different discrete domains.*
- $\forall i \in \{1, 2, ..., q\}$, $\forall h \in Z_i$*, the hybrid trajectory* τ *from* h *(that is,* $\tau(0) = h$*) first reaches* R *at* t_0 *(that is,* $\tau(t_0) \in R$ *and* $\forall t \in [0, t_0[$, $\tau(t) \notin R$*) and the restriction of* τ *on* $[0, t_0]$ *is inside the discrete state* d_s *(that is,* $\forall t \in [0, t_0]$, $\tau(t) \in d_s$*). In this case, we also say that* τ *reaches* $\tau(t_0) \in R$ *inside* d_s.

— *For any hybrid state h that belongs to an input boundary of d_s but does not belong to any Z_i, the hybrid trajectory from h cannot reach any hybrid state of R inside d_s.*

Examples of predecessors in the same discrete state are illustrated in Fig 4.2 where blues rectangles represent R_{target} and blue boxes present their predecessors in the same discrete state.

Proposition 5 *Consider Problem 1, if the hybrid trajectory τ from h_{ini} has already crossed at least one discrete state (we say τ has already crossed a discrete state at t_0 if there exists $t < t_0$ such that $\tau(t_0)$ and $\tau(t)$ do not belong to the same discrete state) without reaching the region R_{target}, then Problem 1 is equivalent to "Does τ reach $Pre_{d_{s_{target}}}(R_{target})$?".*

Proof: According to the Definition 13, we can easily see that if τ reaches $Pre_{d_{s_{target}}}(R_{target})$ then it must reach R_{target}. In case that R_{target} is reached by τ, the only way that $Pre_{d_{s_{target}}}(R_{target})$ is not reached by τ is that h_{ini} is also in the discrete state $d_{s_{target}}$ and τ reaches directly R_{target} inside $d_{s_{target}}$, however since we add the condition "τ from h_{ini} has already crossed at least one discrete state without reaching the region R_{target}", this is not possible. So if R_{target} is reached by τ, then $Pre_{d_{s_{target}}}(R_{target})$ is also reached by τ. □

Based on Proposition 5, we give the following theorem.

Theorem 5 *Problem 1 is decidable if the hybrid trajectory from h_{ini} is a hybrid trajectory attracted by a cycle of discrete domains.*

The idea of Theorem 5 can be explained intuitively by Fig 4.2. For better illustration, the examples in Fig 4.2 has only two dimensions, but this idea can be generalized for n-dimensional models.

In Fig 4.2 left, the hybrid trajectory which reaches hybrid state A, noted by τ, can be considered as two hybrid trajectories: The first one is the part of τ before reaching A and the second one is the part of τ after reaching A. This first one can be considered as a hybrid trajectory halting in finite time so whether it reaches R_{target} is decidable, and in this example it does not reach R_{target}. For the second one, these two following statements can be verified automatically in finite time: 1. The intersection points between this hybrid trajectory and the "right" boundary of discrete state 01 must be located in the

line segment AB (because it converges to the limit cycle with only instant transitions); 2. The line segment AB does not intersect with the predecessor of R_{target} in the same discrete state (the predecessor is illustrated by blue boxes). Based on these two statements, we can prove that this second part cannot reach R_{target} either. In this way, we prove theoretically that R_{target} is not reached by τ, and since this process can be done automatically in finite time, the problem is decidable. Note that in the general case, this "line segment AB" is a (n–1)-dimensional region such that the hybrid trajectory always returns to this region and this region does not intersect the predecessor of R_{target} in the same discrete state.

In Fig 4.2 right, it can be verified automatically in finite time that the limit cycle with only instant transitions (at the center) reaches R_{target}, and that τ converges to this limit cycle, so we can prove that τ finally reaches R_{target}, and this case is thus decidable too.

The formal proof of Theorem 5 is given as follows.

Proof: For a hybrid trajectory τ attracted by a cycle of discrete domains, $\exists t_0$ such that after t_0, τ always stays in a cycle of discrete domains $\mathcal{C}_\mathcal{T} = (\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_p, \mathcal{D}_0)$ and before t_0, τ has already crossed at least one discrete state. τ can be separated on two hybrid trajectories by t_0: The restriction of τ on time interval $[0, t_0]$, noted by $\tau_{[0,t_0]}$, and the restriction of τ on time interval $[t_0, \infty]$, noted by $\tau_{[t_0,\infty]}$.

For the hybrid trajectory $\tau_{[0,t_0]}$, whether it reaches R_{target} is decidable, the proof is the same as the proof of Theorem 4. If it reaches R_{target}, then it is reachable. And if it does not reach R_{target}, then we need to consider $\tau_{[t_0,\infty]}$.

For the hybrid trajectory $\tau_{[t_0,\infty]}$, in case that $\tau_{[0,t_0]}$ does not reach R_{target}, according to Proposition 5, whether $\tau_{[t_0,\infty]}$ reaches R_{target} is equivalent to whether it reaches $Pre_{d_{s_{target}}}(R_{target}) = \bigcup_{i\in\{1,2,...,q\}} Z_i$. If $\forall i \in \{0, 1, 2, ..., p\}$, $\forall j \in \{1, 2, 3, ..., q\}$, $Z_j \not\subseteq \mathcal{D}_i$, then $Pre_{d_{s_{target}}}(R_{target})$ is not reachable. Otherwise, if $\exists i_0 \in \{0, 1, 2, ..., p\}$ and $\exists j_0 \in \{1, 2, 3, ..., q\}$, such that $Z_{j_0} \subseteq \mathcal{D}_{i_0}$, then to verify if Z_{j_0} is reachable, we need to consider the relation between the intersection points of $\tau_{[t_0,\infty]}$ with $\mathcal{D}_{i_0}$ noted by an infinite sequence of hybrid states $(h_{p1}, h_{p2}, ...)$ and Z_{j_0} as follows. When a hybrid trajectory reaches a discrete domain, it might slide inside this discrete domain, in this case, based on the general meaning of "intersection", the intersection points of this hybrid trajectory and this discrete domain are not singular points. In this chapter, we define that the intersection points between a hybrid trajectory and a discrete domain are the points where the hybrid trajectory just reaches the discrete domain.

Since τ always stays inside this cycle of discrete domains, the sequence $(h_{p1}, h_{p2}, ...)$

converges asymptotically to a hybrid state or reaches a hybrid state which belongs to the closure of the compatible zone of $\mathcal{C}_\mathcal{T}$. For both cases, this particular hybrid state is noted by h_∞.

If $h_\infty \in Z_{j_0}$, then $Pre_{d_{s_{target}}}(R_{target})$ is reachable. In case that h_∞ is on the boundary of Z_{j_0} and the sequence $(h_{p1}, h_{p2}, ...)$ converges asymptotically to h_∞ without reaching reaching directly Z_{j_0}, Z_{j_0} is also considered as reachable, which in fact is reached after infinite times of intersections.

If $h_\infty \notin Z_{j_0}$, then we can find a small neighborhood of h_∞ inside $\mathcal{D}_{i_0}$, noted by $\mathcal{N}_{h_\infty}$, such that $\forall h \in \mathcal{N}_{h_\infty}$, $h \notin Z_{j_0}$. Since the sequence $(h_{p1}, h_{p2}, ...)$ converges to h_∞ or finally reaches h_∞, we can find n_0 such that $\forall n > n_0$, $h_{pn} \in \mathcal{N}_{h_\infty}$. This means that after that $\tau_{[t_0,\infty]}$ reaches h_{pn_0}, it will never reach Z_{j_0}. So, in this case, to verify if Z_{j_0} is reached, we only need to verify if the finite sequence $(h_{p1}, h_{p2}, ..., h_{pn_0})$ ever reaches Z_{j_0}, which is decidable.

Now we see that, for a certain Z_i, whether $\tau_{[t_0,\infty]}$ reaches Z_i or not is decidable. Since $Pre_{d_{s_{target}}}(R_{target})$ has a finite number of such Z_i, whether $\tau_{[t_0,\infty]}$ reaches $Pre_{d_{s_{target}}}(R_{target})$ is also decidable. □

From the above proof of Theorem 5, we can derive a method to check the reachability of hybrid trajectories attracted by cycles of discrete domains, which is introduced later in Section 4.3.

We also develop the following theorem to determine automatically whether a hybrid trajectory is attracted by a cycle of discrete domains. In order to simplify this theorem, for the cycle of discrete domains $\mathcal{C}_\mathcal{T} = (\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_p, \mathcal{D}_0)$ and the hybrid state h_0 considered in this theorem, we note that:

— The reduction matrix and the constant vector of $\mathcal{C}_\mathcal{T}$ are A and b respectively (the reduction matrix and the constant vector are defined in Section 3.3.2).

— The reduction compatible zone of $\mathcal{C}_\mathcal{T}$ is described by linear constraints $\{x \mid Wx > c\}$ where c is a vector and W is a matrix. W is of size $n_0 \times n_1$, where n_1 is the number of dimensions of $r(h_0)$ ($r(h_0)$ is the reduction vector of h_0 which is defined in Section 3.3.2). W_i is the i^{th} line of matrix W (W_i is of size $1 \times n_1$) and c_i is the i^{th} component of vector c.

— $r_\infty = \lim_{n\to\infty} f^n(r(h_0))$ where $f(x) = Ax + b$.

— The eigenvalues and eigenvectors of A are $\{\lambda_i \mid i \in \{1, 2, ..., n_1\}\}$ and $\{v_i \mid i \in \{1, 2, ..., n_1\}\}$ respectively. λ_1 is chosen as the eigenvalue with the maximum absolute value among the eigenvalues that differ from 1.

— The decomposition of $r(h_0) - r_\infty$ in the directions of eigenvectors of the reduction matrix A is noted as $r(h_0) - r_\infty = \sum_{i=1}^{n_1} \alpha_i v_i$.

Theorem 6 *A hybrid trajectory τ is attracted by a cycle of discrete domains if and only if τ reaches h_0 which belongs to the compatible zone of a cycle of discrete domains $\mathcal{C}_\mathcal{T} = (\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_p, \mathcal{D}_0)$ such that $\mathcal{D}_0$ has no free dimension (meaning that, in $\mathcal{D}_0$, boundaries are reached in all dimensions) or the following conditions are satisfied.*

— *$\mathcal{D}_0$ has at least one free dimension.*

— *$\forall i \in \{1, 2, ..., n_1\}, |\lambda_i| \leq 1 \wedge \lambda_i \neq -1$.*

— *$\forall i \in \{1, 2, ..., n_0\}$, we have either $W_i r_\infty = c_i$ or $W_i r_\infty > c_i$. We use I_e to represent the maximum set of integers such that $\forall i \in I_e, W_i r_\infty = c_i$ and we use I_n to represent the maximum set of integers such that $\forall i \in I_n, W_i r_\infty > c_i$.*

— *If $\lambda_1 \neq 0$ (we assume that λ_1 is unique if $\lambda_1 \neq 0$) and I_e is not empty, then λ_1 is positive.*

— *If $\lambda_1 \neq 0$, then $\forall i \in I_e, \forall j \in \{2, ..., n_1\}, |W_i v_1 \alpha_1| > n_1 |W_i v_j \alpha_j|$ (we ignore the case where $\exists i \in I_e, W_i v_1 = 0$).*

— *If $\lambda_1 \neq 0$, then $\forall i \in I_n, \max_{\beta \in \{-1,1\}^{n_1}} \|\sum_{j=1}^{n_1} \beta_j \alpha_j v_j\|_2 < \frac{W_i r_\infty - c_i}{\|W_i\|_2}$.*

The main idea of Theorem 6 is illustrated in Fig 4.3 where the huge rectangle represents a discrete domain $\mathcal{D}$ which has two free dimensions and the zone surrounded by dashed lines represents the compatible zone $\mathcal{S}$ (which is a open set) of a certain cycle of discrete domains $\mathcal{C}_\mathcal{T}$. Each dashed line l_{ci} represents a linear constraint of the form $w^T x > c$ where w, x are vectors and c is a real number. The fact that a hybrid trajectory τ is attracted by $\mathcal{C}_\mathcal{T}$ is equivalent to the fact that the intersection points between τ and $\mathcal{D}$, noted by the sequence $(h_1, h_2, ...)$, always stay inside $\mathcal{S}$ and converge to $(\lambda_1 \neq 0)$ or reach $(\lambda_1 = 0)$ h_∞, which belongs to the closure of $\mathcal{S}$. This idea of using the intersection points between a hybrid trajectory and a hyperplan to study the properties of this hybrid trajectory is, like the previous chapter, also based on Poincaré map.

Whether h_∞ belongs to the closure of $\mathcal{S}$ or not can be easily verified by using these linear constraints. A necessary condition for this sequence to always satisfy these linear constraints is that the absolute values of all eigenvalues of the reduction matrix of $\mathcal{C}_\mathcal{T}$ are less than or equal to 1. In case that these eigenvalues satisfy this necessary condition, to verify if this sequence always satisfies these linear constraints, we separate these constraints on two classes: The first class contains all constraints which are not reached by h_∞: l_{c2}, l_{c3}, l_{c4}, the second class contains all constraints which are reached by h_∞: l_{c1}, l_{c5}. To verify if l_{c2}, l_{c3}, l_{c4} are always satisfied, we can verify if this sequence enters and stays in a circle

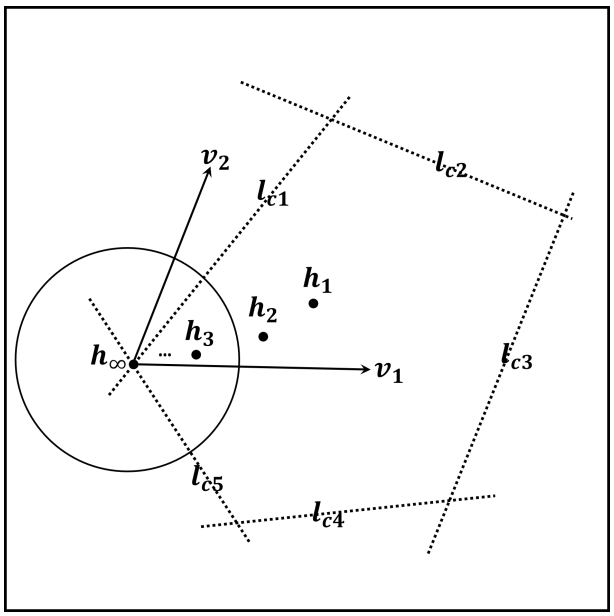


Figure 4.3 – Illustration of the idea of Theorem 6. The surrounding rectangle represents a discrete domain which has two free dimensions and the zone surrounded by dashed lines represents the compatible zone of a certain cycle of discrete domains. v_1 and v_2 represent the two eigenvectors of the reduction matrix of this cycle of discrete domains. $h_1, h_2, ...$ represent the intersection points between a hybrid trajectory and this discrete domain.

centered by h_∞ which only contains hybrid states satisfying constraints l_{c2}, l_{c3}, l_{c4} (this is related to the condition: If $\lambda_1 \neq 0$, then $\forall i \in I_n, \max_{\beta\in\{-1,1\}^{n_1}} \|\sum_{j=1}^{n_1} \beta_j \alpha_j v_j\|_2 < \frac{W_i r_\infty - c_i}{\|W_i\|_2}$). Such circle can always be found if it is sufficiently small, for example: the circle in Fig 4.3. To verify if l_{c1}, l_{c5} are always satisfied, we can verify if this sequence is sufficiently "close" to v_1 which is the eigenvector related to the eigenvalue with the maximum absolute value among the eigenvalues that differ from 1 and which also "points into" $\mathcal{S}$ (this is related to the condition: If $\lambda_1 \neq 0$, then $\forall i \in I_e, \forall j \in \{2, ..., n_1\}, |W_i v_1 \alpha_1| > n_1 |W_i v_j \alpha_j|$). Here, sufficiently "close" to v_1 means intuitively that the angle between $\overrightarrow{h_\infty h_i}$ and v_1 is sufficiently small.

The formal proof of Theorem 6 is given as follows.

Proof: The proof of the sufficiency part:

In the case that a hybrid trajectory τ is attracted by a cycle of discrete domains, then after certain moment, τ will always stay in the same cycle of discrete domains, noted by $\mathcal{C}_\mathcal{T} = (\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_p, \mathcal{D}_0)$.

We consider, at first, the case that there is one $\mathcal{D}_i$ of $\mathcal{C}_\mathcal{T}$ such that $\mathcal{D}_i$ has at least one

free dimension (meaning that there exists one dimension such that the boundaries are not reached in this dimension), without loss of generality we assume that $\mathcal{D}_0$ has at least one free dimension, so the first condition is satisfied.

A priori, we cannot ensure the second condition, because the decomposition of $r(h_0) - r_\infty$ in the directions of certain eigenvectors of the reduction matrix of $\mathcal{C}_\mathcal{T}$ can be zero and these eigenvectors happen to be associated to eigenvalues of which the absolute values are greater than 1, however this is extremely rare. So in this chapter we ignore this case, and based on this assumption, we must have that for any eigenvalue λ_i, $|\lambda_i| \leq 1$, otherwise we cannot ensure that τ always stays inside this cycle, because if there exists an eigenvalue such that $|\lambda_i| > 1$, then r_∞ is infinitely far away from $r(h_0)$. If there exists $\lambda_i = -1$, we can double this cycle of discrete domains (the doubled cycle of discrete domains is $(\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_p, \mathcal{D}_0, \mathcal{D}_1, ..., \mathcal{D}_p, \mathcal{D}_0)$) to eliminate this eigenvalue. So the second condition is satisfied.

The intersection points of τ with $\mathcal{D}_0$ will converge to or reach a certain hybrid state $h_\infty = (\pi_\infty, d_{s_0})$ which belongs to the closure of the compatible zone. We have that r_∞ is the short version of π_∞ by only considering the dimensions where boundaries are not reached in $\mathcal{D}_0$. Based on Assumption 5, the reduction compatible zone can be described by $\{x \mid Wx > c\}$. If h_∞ belongs to the compatible zone, then we have $\forall i \in \{1, 2, ..., n_0\}$, $W_i r_\infty > c_i$. Else if h_∞ does not belong to the compatible zone but belongs to the closure of the compatible zone, then we have that for some $i \in \{1, 2, ..., n_0\}$, $W_i r_\infty = c_i$. This means that some "boundaries" of the reduction compatible zone are reached, and for others $i \in \{1, 2, ..., n_0\}$, we have $W_i r_\infty > c_i$. So the third condition is satisfied.

We note the intersection hybrid states of τ and $\mathcal{D}_0$ as the sequence $(h_0, h_1, h_2, ...)$. We have that $r(h_n) = r_\infty + \sum_{i=1}^{n_1} \lambda_i^n \alpha_i v_i$ where v_i are eigenvectors of the reduction matrix of $\mathcal{C}_\mathcal{T}$ and $r(h_0) - r_\infty = \sum_{i=1}^{n_1} \alpha_i v_i$. By choosing λ_1 as the eigenvalue with maximum absolute eigenvalue among the eigenvalues that differ from 1, if $\lambda_1 \neq 0$, then we have: $\lim_{n\to\infty} |\lambda_1^n \alpha_1| \gg |\lambda_i^n \alpha_i|$ where $i \neq 1$, and $\lim_{n\to\infty} \sum_{i=1}^{n_1} \lambda_i^n \alpha_i v_i \to 0$. This proves that the fifth and the sixth conditions are satisfied.

Suppose that $\lambda_1 \neq 0$. Since the sequence $(h_0, h_1, h_2, ...)$ is inside the compatible zone, if I_e is not empty, then for any $j \in I_e$, we should have $\forall n \in \mathbb{N}, W_j \sum_{i=1}^{n_1} \lambda_i^n \alpha_i v_i > 0$. When n is sufficiently big, for any $j \in I_e$, the sign of $W_j \sum_{i=1}^{n_1} \lambda_i^n \alpha_i v_i$ is dominated only by $W_j \lambda_1^n \alpha_1 v_1$, meaning that $W_j \sum_{i=1}^{n_1} \lambda_i^n \alpha_i v_i$ and $W_j \lambda_1^n \alpha_1 v_1$ have the same sign, in fact, when n is sufficiently big, we have $|W_j \lambda_1^n \alpha_1 v_1| > \sum_{i=2}^{n_1} |W_j \lambda_i^n \alpha_i v_i|$, this means that adding $\sum_{i=2}^{n_1} W_j \lambda_i^n \alpha_i v_i$ to $W_j \lambda_1^n \alpha_1 v_1$ does not change the sign of $W_j \lambda_1^n \alpha_1 v_1$. So the sign of λ_1 is

positive, otherwise we can not ensure that $\forall j \in I_e, \forall n \in \mathbb{N}, W_j \sum_{i=1}^{n_1} \lambda_i^n \alpha_i v_i > 0$, which is equivalent to $\forall j \in I_e, \forall n \in \mathbb{N}, W_j \lambda_1^n \alpha_1 v_1 > 0$ when n is sufficiently big. This proves that the fourth condition is satisfied.

In the case that any $\mathcal{D}_i$ of $\mathcal{C}_\mathcal{T}$ has no free dimension, we can easily prove the sufficiency part.

The proof of the necessity part:

If $\mathcal{D}_0$ has no free dimension, then τ reaches a periodic hybrid trajectory which is a special case of τ being attracted by a cycle of discrete domains.

If these six conditions are satisfied, to prove that τ is attracted by a cycle of discrete domains, we only need to prove that τ always returns to the compatible zone of $\mathcal{C}_\mathcal{T}$. This can be easily proved if $\lambda_1 = 0$. From now on, we suppose $\lambda_1 \neq 0$. We note that the hybrid trajectory from h_0 returns to $\mathcal{D}_0$ at h_1. We have that $r(h_1) - r_\infty = \sum_{i=1}^{n_1} \lambda_i \alpha_i v_i$. A sufficient condition for "h_0 belongs to the compatible zone" is $(\forall j \in I_e, W_j \sum_{i=1}^{n_1} \alpha_i v_i > 0) \wedge (\forall j \in I_n, \max_{\beta_i \in \{-1,1\}} \| \sum_{i=1}^{n_1} \beta_i \alpha_i v_i \|_2 < \frac{W_j r_\infty - c_i}{\|W_j\|_2})$ which is satisfied in this case. In fact, the first part of this sufficient condition is a necessary condition for h_0 to belong to the compatible zone and the second part is ensured by the sixth condition. Now we want to prove that this sufficient condition is also satisfied for h_1. The fifth condition indicates that for any $j \in I_e$ the sign of $W_j \sum_{i=1}^{n_1} \alpha_i v_i$ is dominated by the sign of $W_j \alpha_1 v_1$, so $W_j \alpha_1 v_1$ is also positive. Since $\forall i \neq 1$, we have either $|\lambda_1| > |\lambda_i|$ or $|\lambda_1| < |\lambda_i| \wedge \alpha_i = 0$, so we have $\forall j \in I_e, \forall i \in \{2, ..., n_1\}, |W_j v_1 \lambda_1 \alpha_1| > n_1 |W_j v_i \lambda_i \alpha_i|$. This means that for any $j \in I_e$ the sign of $W_j \sum_{i=1}^{n_1} \alpha_i \lambda_i v_i$ is also dominated by the sign of $W_j \alpha_1 \lambda_1 v_1$. Since λ_1 is positive if I_e is not empty, we have $\forall j \in I_e, W_j \sum_{i=1}^{n_1} \lambda_i \alpha_i v_i > 0$, because the sign of $W_j \sum_{i=1}^{n_1} \lambda_i \alpha_i v_i$ is same as the sign of $W_j \alpha_1 \lambda_1 v_1$ which is same as the sign of $W_j \alpha_1 v_1$ of which the sign is positive. Since $\forall \lambda_i, |\lambda_i| \leq 1$, we have $\max_{\beta_i \in \{-1,1\}} \| \sum_{i=1}^{n_1} \beta_i \lambda_i \alpha_i v_i \|_2 \leq \max_{\beta_i \in \{-1,1\}} \| \sum_{i=1}^{n_1} \beta_i \alpha_i v_i \|_2$. So we also have $\forall j \in I_n, \max_{\beta_i \in \{-1,1\}} \| \sum_{i=1}^{n_1} \beta_i \lambda_i \alpha_i v_i \|_2 < \frac{W_j r_\infty - c_i}{\|W_j\|_2}$. By now we proved that h_1 also belongs to the compatible zone. By mathematical induction, we can prove that τ always returns to the compatible zone of $\mathcal{C}_\mathcal{T}$. So τ is attracted by a cycle of discrete domains. □

Theorem 6 is used later in the algorithm to analyze reachability.

4.2.3 Chaotic Hybrid Trajectories

In this thesis, a hybrid trajectory of HGRN is called a chaotic hybrid trajectory if it does not reach a fixed point and it is not attracted by a cycle of discrete domains. So all hybrid trajectories which are not included in the previous two classes are considered

chaotic hybrid trajectories. It is worth mentioning that the dynamics of chaotic hybrid trajectories, a priori, can be different from the chaotic dynamics of classic nonlinear dynamical systems. The reason why we still use the terminology "chaotic" is that a similar concept of chaos has been used in some pre-existing works of other hybrid systems [55, 56].

As stated before, the sequence of discrete domains reached by a hybrid trajectory of the previous two classes is of the form: $(\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m)$ or $(\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_k, (\mathcal{D}_{k+1}, \mathcal{D}_{k+2}, ..., \mathcal{D}_{k+p})_n)$. While a chaotic hybrid trajectory corresponds to an irregular sequence of discrete domains, like an irrational number with each digit replaced by a discrete domain.

To prove such chaotic hybrid trajectories exist, we have constructed a HGRN with chaotic hybrid trajectories based on a pre-existing model of electrical circuit with a chaotic attractor [56].

In our work, we have not yet found a method to check reachability for chaotic hybrid trajectories, which, a priori, can be undecidable. So, in this subsection, we only introduce a method to predict whether a hybrid trajectory is chaotic, which is based on a necessary condition.

We prove that a chaotic hybrid trajectory has the following properties.

Property 1 *For a chaotic hybrid trajectory τ, there exist t_0 and a finite set of discrete domains L_D, such that after t_0, τ cannot reach any discrete domain which does not belong to L_D, and for any discrete domain $\mathcal{D}_0 \in L_D$, $\mathcal{D}_0$ is reached by τ an infinite number of times.*

Proof: This can be proved by the fact that the number of discrete domains is finite and the hybrid trajectory does not stay in a particular discrete domain. In fact, this property is true for any hybrid trajectory which is not a hybrid trajectory halting in finite time. □
Following Property 1, for any $\mathcal{D}_0 \in L_D$, τ will return to $\mathcal{D}_0$ an infinite number of times, and each time it stays inside a sequence of discrete domains of the form $(\mathcal{D}_0, ..., \mathcal{D}_0)$ (only the first and the last discrete domain are $\mathcal{D}_0$). The set of all such sequences of discrete domains is noted by $L_{\mathcal{D}_0}$, and we make the following assumption.

Assumption 6 *$L_{\mathcal{D}_0}$ is a finite set.*

This Assumption is based on the fact that the number of discrete domains is limited and the dynamics in the discrete states is simple (a constant vector).

Property 2 *Following the notions of Property 1, for any $\mathcal{D}_0 \in L_D$, we can find $t_1 > t_0$ such that τ reaches $\mathcal{D}_0$ at t_1 and there exists a finite set of sequences of discrete domains L_T which satisfies the following properties:*

- *— $\forall \mathcal{T} \in L_T$, $\mathcal{T}$ is of the form $(\mathcal{D}_0, ..., \mathcal{D}_0)$.*
- *— From t_1, τ will return to $\mathcal{D}_0$ an infinite number of times, and each time it stays in a sequence of discrete domains which belongs to L_T. More formally, if τ reaches $\mathcal{D}_0$ at t_α, then from t_α, it will return to $\mathcal{D}_0$ at t_β, and the restriction of τ on $[t_\alpha, t_\beta]$ is inside a sequence of discrete domains of L_T.*
- *— From t_1, $\forall \mathcal{T} \in L_T$, $\mathcal{T}$ is crossed by τ an infinite number of times.*
- *— L_T contains at least two sequences of discrete domains.*

Proof: For the first two properties, according to Property 1, from t_1, τ will return to $\mathcal{D}_0$ an infinite number of times, and each time it stays in a sequence of discrete domains of the form $(\mathcal{D}_0, ..., \mathcal{D}_0)$, all these sequence constitute L_T. And according to Assumption 6, L_T contains a finite number of sequences.

For the third property, for any $\mathcal{T} \in L_T$, if it is crossed by τ a finite number of times, then we can increase the value of t_1 such that from t_1, $\mathcal{T}$ is not crossed by τ any more. This result is also based on the fact that L_T is a finite set which is a result of Assumption 6.

For the fourth property, if there is only one sequence of discrete domains, then after certain moment, τ will always stay in the same cycle of discrete domains. In this case, τ is attracted by a cycle of discrete domains and is not a chaotic hybrid trajectory. □

Following the notations of Property 2, the sequence of discrete domains crossed by τ from t_1 can be described by the sequence $(\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, ...)$, where $\forall i \in \mathbb{N}, \mathcal{T}_i \in L_T$. This sequence has the following properties.

Property 3 *$\exists i \in \mathbb{N}$, such that $\mathcal{T}_i \neq \mathcal{T}_{i+1}$.*

Proof: This is a direct result of the fact that all elements of L_T must appear in the sequence $(\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, ...)$ and L_T has at least two elements. □

Property 4 *$\forall i \in \mathbb{N}, \exists k \in \mathbb{N}$, such that $\mathcal{T}_i = \mathcal{T}_{i+k}$.*

Proof: If it is not true, then it means that $\mathcal{T}_i$ is crossed by τ for finite times, which contradicts with Property 2. □

By combining Property 3 and Property 4, we get a necessary condition for τ to be a chaotic hybrid trajectory:

Property 5 *$\exists i \in \mathbb{N}, \exists k \in \mathbb{N}, k \neq 1$, such that $\mathcal{T}_i \neq \mathcal{T}_{i+1}$ and $\mathcal{T}_i = \mathcal{T}_{i+k}$.*

This condition is used in the following section to predict whether a hybrid trajectory is a chaotic hybrid trajectory.

4.3 Reachability Analysis Algorithm

In this section, we present our reachability analysis algorithm. The main algorithm is presented in Algorithm 1. Its main idea is that we continue simulating the hybrid trajectory τ from h_{ini} until reaching one of the following situations:

— τ reaches R_{target}. In this case, the algorithm stops and returns "R_{target} is reached".

— At some time t_0, τ is already attracted by a cycle of discrete domains and the periodic hybrid trajectory, which attracts τ, reaches R_{target}. In this case, the algorithm stops and returns "R_{target} is reached".

— At some time t_0, τ is already attracted by a cycle of discrete domains and after t_0 there is no more chance to reach R_{target}. In this case, the algorithm stops and returns "R_{target} is not reached".

— Property 5 is satisfied, therefore τ is probably chaotic. In this case, the algorithm stops and returns "unknown result".

More details about Algorithm 1 are discussed as follows.

To check if a transition $h \rightarrow h'$ reaches $R_{target} = \left\{(\pi, d_{s_{target}}) \mid \pi^i \in [a_i, b_i], i \in \{1, 2, ..., N\}\right\}$, we need to consider two cases: Firstly, if $h \rightarrow h'$ is an instant transition, then we only need to verify if h' belongs to R_{target}; secondly, if $h \rightarrow h'$ is a non-instant transition, meaning that $h \rightarrow h'$ is an N-dimensional straight line, then we only need to verify if $h \rightarrow h'$ crosses a boundary of R_{target}. To give an example of the second case, if a boundary of R_{target} is $\left\{(\pi, d_{s_{target}}) \mid \pi^{i_0} = a_{i_0}, \pi^i \in [a_i, b_i], i \in \{1, 2, ..., N\} \setminus \{i_0\}\right\}$, we only need to verify if $h \rightarrow h'$ contains a hybrid state $h_t = (\pi_t, d_{s_{target}})$ such that $\pi_t^{i_0} = a_{i_0}$ and $\forall i \in \{1, 2, ..., N\} \setminus \{i_0\}, \pi_t^i \in [a_i, b_i]$.

To determine if the current simulation is attracted by a cycle of discrete domains or if the current simulation is probably a chaotic hybrid trajectory, we use Theorem 6 or Property 5 respectively.

The main idea of the function *Stop_condition* is the following. Its second argument, $Cycle_{\mathcal{D}}$, represents the cycle of discrete domains attracting the current hybrid trajectory. Its first argument, $Cycle_h$, represents the last part of the simulation inside $Cycle_{\mathcal{D}}$. The

Algorithm 1 Reachability analysis algorithm

Input 1: A hybrid state $h_{ini} = (\pi_{ini}, d_{s_{ini}})$
Input 2: A region $R_{target} = \left\{(\pi, d_{s_{target}}) \mid \pi^i \in [a_i, b_i], i \in \{1, 2, ..., N\}\right\}$
Output: "R_{target} is reached", "R_{target} is not reached" or "unknown result"

Current hybrid state $h := h_{ini}$
while h is not a fixed point **do**
    $h' :=$ next hybrid state so that $h \rightarrow h'$ is a transition
    if Transition $h \rightarrow h'$ reaches R_{target} **then**
        return "R_{target} is reached"
    else
        $h := h'$
        if Current simulation is attracted by a cycle of discrete domains **then**
            if $Stop\_condition(Cycle_h, Cycle_{\mathcal{D}}, R_{target})$ returns "Yes" **then**
                return "R_{target} is not reached"
            else if $Stop\_condition(Cycle_h, Cycle_{\mathcal{D}}, R_{target})$ returns "Reached" **then**
                return "R_{target} is reached"
            end if
        else if Current simulation is probably a chaotic hybrid trajectory **then**
            return "unknown result"
        end if
    end if
end while
return "R_{target} is not reached"

objective of the function *Stop_condition* is, knowing that this hybrid trajectory is attracted by a cycle of discrete domains, to determine if the hybrid trajectory can reach R_{target} after an infinite number of transitions (see Fig 4.2 right). If it is the case, the function returns "Reached". Otherwise, if from the current hybrid state, there is no more chance to reach R_{target} (see Fig 4.2 left), then the function returns "Yes". For both cases, this function can give the right answer in finite time, and the result stops the algorithm. However, if both cases don't apply, the function returns "No" and the algorithm continues. The idea of the function *Stop_condition* is based on the proof of Theorem 5. The details of the function *Stop_condition* are presented in Algorithm 2.

Algorithm 2 Stop condition

Input 1: A list of hybrid states $Cycle_h = (h_0, h_1, h_2, ..., h_p, h'_0)$ which describes the last part of the simulation inside $Cycle_{\mathcal{D}}$

Input 2: A list of discrete domains $Cycle_{\mathcal{D}} = (\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_p, \mathcal{D}_0)$ such that $\forall i \in \{0, 1, 2, ..., p\}, h_i \in \mathcal{D}_i, h'_0 \in \mathcal{D}_0$ and the hybrid trajectory from h_0 is attracted by $(\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_p, \mathcal{D}_0)$

Input 3: A region $R_{target} = \left\{(\pi, d_{s_{target}}) \mid \pi^i \in [a_i, b_i], i \in \{1, 2, ..., N\}\right\}$

Output (three possibilities): "Yes", "No", and "Reached"

1: Compute the hybrid states $h_0^\infty, h_1^\infty, h_2^\infty, ..., h_p^\infty$ such that the intersection hybrid states between this hybrid trajectory and $\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_p$ converge to or reach these hybrid states after an infinite number of transitions.
2: $var\_stop \leftarrow True$
3: **for** $i \in \{0, 1, 2, ..., p\}$ **do**
4: **if** $\mathcal{D}_i$ is on an input boundary of $d_{s_{target}}$ **then**
5: **if** Hybrid trajectory from h_i^∞ reaches R_{target} inside $d_{s_{target}}$ **then**
6: **return** "Reached"
7: **else if** Non-intersection condition is satisfied **then**
8: $var\_stop \leftarrow var\_stop \wedge True$
9: **else if** Non-intersection condition is not satisfied **then**
10: $var\_stop \leftarrow var\_stop \wedge False$
11: **end if**
12: **end if**
13: **end for**
14: **if** $var\_stop = True$ **then**
15: **return** "Yes"
16: **else if** $var\_stop = False$ **then**
17: **return** "No"
18: **end if**

In order to describe the Non-intersection condition in Algorithm 2, we note the prede-

cessor in the same discrete state of R_{target} as $Pre_{d_{s_{target}}}(R_{target}) = \bigcup_{i\in\{1,2,...,q\}} Z_i$. We note $r(Z_i)$ as the set of the reduction vectors of all hybrid states of Z_i.

This Non-intersection condition (its formal definition will be given later) is a sufficient condition for that $\forall j \in \{1,2,...,q\}$, if $Z_j \subset \mathcal{D}_i$ ($\mathcal{D}_i$ is a discrete domain that belongs to the cycle of discrete domains attracting the hybrid trajectory), then $\left\{x \in \mathcal{S}_r(\mathcal{D}_i) \mid \|x - r(h_i^\infty)\|_2 \leq \max_{\beta\in\{-1,1\}^{n_1}} \|\sum_{k=1}^{n_1} \beta_k\alpha_k v_k\|_2\right\} \cap r(Z_j) = \emptyset$ where $\mathcal{S}_r(\mathcal{D}_i)$ is the reduction compatible zone of the cycle of discrete domains $(\mathcal{D}_i, \mathcal{D}_{i+1}, ..., \mathcal{D}_p, \mathcal{D}_0, ..., \mathcal{D}_i)$, n_1 is the dimension of $\mathcal{S}_r(\mathcal{D}_i)$, $r(h_i) - r(h_i^\infty) = \sum_{k=1}^{n_1} \alpha_k v_k$, v_k are the eigenvectors of the reduction matrix of this cycle $(\mathcal{D}_i, \mathcal{D}_{i+1}, ..., \mathcal{D}_p, \mathcal{D}_0, ..., \mathcal{D}_i)$.

The idea of the Non-intersection condition is illustrated in Fig 4.4 by a 2-dimensional example. Initially, we want to verify $\forall j \in \{1,2,...,q\}$, if $Z_j \subset \mathcal{D}_i$, then $\left\{x \in \mathcal{S}_r(\mathcal{D}_i) \mid \|x - r(h_i^\infty)\|_2 \leq \max_{\beta\in\{-1,1\}^{n_1}} \|\sum_{k=1}^{n_1} \beta_k\alpha_k v_k\|_2\right\} \cap r(Z_j) = \emptyset$. This is equivalent to say that the black box does not intersect the blue boxes in Fig 4.4 left. The reason why we want to verify this condition, noted here as Condition1, is that once it is satisfied, there is no more chance that a hybrid trajectory can reach R_{target} directly from $\mathcal{D}_i$ (here "directly" means that it does not reach another discrete state before reaching R_{target}). However, it is complicated to compute directly the intersection of these sets. So, firstly we overestimate the set of all hybrid states in this discrete state which can be reached by hybrid trajectories from $\left\{h \in \mathcal{S}(\mathcal{D}_i) \mid \|r(h) - r(h_i^\infty)\|_2 \leq \max_{\beta\in\{-1,1\}^{n_1}} \|\sum_{k=1}^{n_1} \beta_k\alpha_k v_k\|_2\right\}$ and which are reached before these hybrid trajectories reaching new discrete states. This set of hybrid states is illustrated by the red zone in Fig 4.4 middle (note that Fig 4.4 is just a illustration, the shape of the real red zone is slightly different from the one in this figure). If this red zone does not intersect the blue rectangle, then Condition1 is satisfied. However, this intersection is still difficult to compute. So secondly, we move the "thickness" of the red zone to the blue rectangle in Fig 4.4 right. Now to verify if Condition1 is satisfied, we only need to verify if the red trajectory in Fig 4.4 right reaches the blue and red zone, which can be done automatically and is called Non-intersection condition.

The Non-intersection condition is defined formally as follows. The hybrid trajectory from h_i^∞ can be represented by $h_i^\infty = h_{i,0}^\infty \to h_{i,1}^\infty \to ... \to h_{i,K}^\infty$ where $\left\{h_{i,j}^\infty, j \in \{1,2,...,K\}\right\}$ represents all hybrid states at which this hybrid trajectory reaches new boundaries (similar to the turning states in Chapter 3) and $h_{i,K}^\infty$ represents the hybrid state at which this hybrid trajectory reaches an output boundary of $d_{s_{target}}$ for the first time. We note $u_j, j \in \{1,2,...,K\}$ as the left derivative of hybrid state $h_{i,j}^\infty$. We use $w_j \in \{0,1\}^N, j \in \{1,2,...,K\}$ to describe the dimension where the new boundary is reached

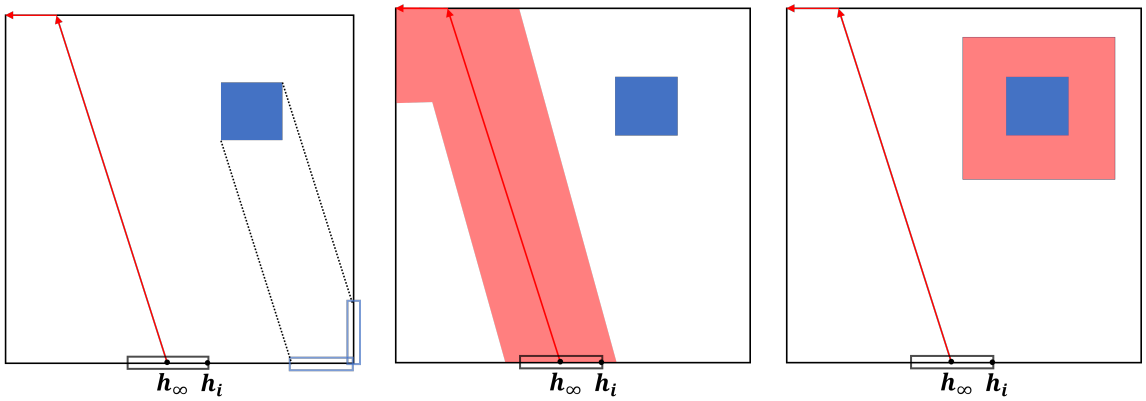


Figure 4.4 – Illustration of the Non-intersection condition in Algorithm 2.

in each transition, for the transition $h^{\infty}_{i,j-1} \rightarrow h^{\infty}_{i,j}, j \in \{1,2,...,K\}$: if the new boundary is reached in the $m_j{}^{\text{th}}$ dimension, then $w_j^{m_j} = 1$ and $w_j^k = 0, k \in \{1,2,...,N\} \setminus \{m_j\}$. Based on the notations of u_j and w_j, we compute a sequence $l_j, j \in \{0,1,2,...,K-1,K\}$ as follows: $l_0 = \max_{\beta\in\{-1,1\}^{n_1}} \|\sum_{k=1}^{n_1} \beta_k \alpha_k v_k\|_2$ and $\forall j \in \{1,2,...,K-1,K\}, l_j = \frac{l_{j-1}}{|\frac{u_j \cdot w_j}{\|u_j\|_2}|}$ where $u_j \cdot w_j$ is the inner product of u_j and w_j. The Non-intersection condition is satisfied if the hybrid trajectory from $h^{\infty}_{i,0}$ does not reach directly the region $R'_{target} = \left\{(\pi, d_{s_{target}}) \mid \pi^i \in [\max(a_i - l_K, 0), \min(b_i + l_K, 1)], i \in \{1,2,...,N\}\right\}$. The reason why we compute this sequence l_j is illustrated in Fig 4.5 by a 2-dimensional example. Intuitively, when a bunch of hybrid trajectories with radius l_{j-1} reaches a new boundary, the radius can increase. We can see that the sequence l_j is increasing. To simplify the computation, we choose the maximum value l_K to overestimate the hybrid states that can be reached by these hybrid trajectories. This choice of the value l_K is only for overestimation, it is possible that a smaller value also works.

Here we prove that the Non-intersection condition is a sufficient condition so that $\forall j \in \{1,2,...,q\}$, if $Z_j \subset \mathcal{D}_i$, then $\left\{x \in \mathcal{S}_r(\mathcal{D}_i) \mid \|x - r(h_i^{\infty})\|_2 \leq \max_{\beta\in\{-1,1\}^{n_1}} \|\sum_{k=1}^{n_1} \beta_k \alpha_k v_k\|_2\right\} \cap r(Z_j) = \emptyset$. We note this hybrid trajectory $h_i^{\infty} = h^{\infty}_{i,0} \rightarrow h^{\infty}_{i,1} \rightarrow ... \rightarrow h^{\infty}_{i,K}$ which begins from $h^{\infty}_{i,0}$ and finally reaches $h^{\infty}_{i,K}$ as τ_i. If the Non-intersection condition is satisfied, then for any hybrid state h_x which belongs to R_{target} and for any hybrid state h_y which belongs to τ_i, the distance between h_x and h_y is greater than l_K. On the other hand, for any hybrid trajectory τ_x which is inside this discrete state and begins from the region $\left\{h \in \mathcal{S}(\mathcal{D}_i) \mid \|r(h) - r(h_i^{\infty})\|_2 \leq \max_{\beta\in\{-1,1\}^{n_1}} \|\sum_{k=1}^{n_1} \beta_k \alpha_k v_k\|_2\right\}$ where $\mathcal{S}(\mathcal{D}_i)$ is the compatible zone of this cycle, for any hybrid state h'_x on τ_x, we can find a hybrid

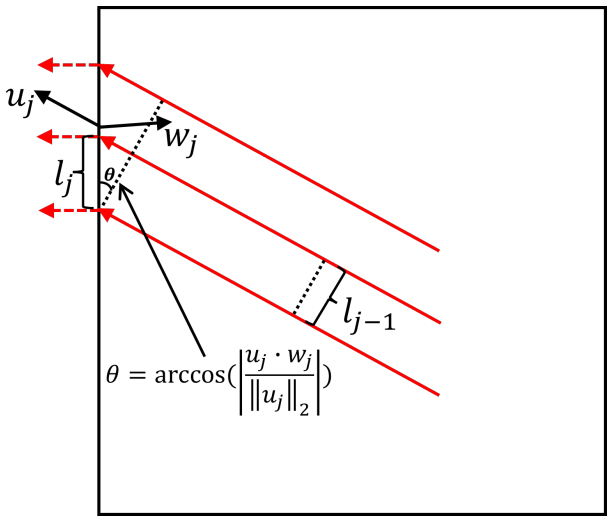


Figure 4.5 – Illustration of the idea of computing the sequence l_j.

state h'_y on τ_i, such that the distance between h'_x and h'_y is smaller than l_K. If τ_x crosses R_{target}, then there exists a hybrid state h''_y which is on τ_x and also belongs to R_{target}, such that $\exists h''_x$ on τ_i, the distance between h''_x and h''_y is smaller than l_K which is not possible as h''_y belongs to R_{target}. This proves that $\forall j \in \{1, 2, ..., q\}$, if $Z_j \subset \mathcal{D}_i$, then $\left\{x \in \mathcal{S}_r(\mathcal{D}_i) \mid \|x - r(h_i^\infty)\|_2 \leq \max_{\beta \in \{-1,1\}^{n_1}} \|\sum_{k=1}^{n_1} \beta_k \alpha_k v_k\|_2\right\} \cap r(Z_j) = \emptyset$.

In fact, if the hybrid trajectory from h_{ini} does not reach R_{target} and h_i is sufficiently close to h_i^∞, then the Non-intersection condition must be satisfied, because L_K is sufficiently close to 0 in this case. Based on this fact, it can be proved that if the hybrid trajectory from h_{ini} is attracted by a cycle of discrete domains and does not reach R_{target}, then the Non-intersection condition must be verified in finite time for all discrete domains which belong to an input boundary of the discrete state of R_{target} and belong to the attractive cycle of discrete domains (the cycle of discrete domains which attracts the hybrid trajectory from h_{ini}). In this case, this algorithm returns "Yes", meaning that the hybrid trajectory from h_{ini} has no more chance to reach R_{target}.

It can be proved that Algorithm 1 always stops in finite time. Firstly, if the hybrid trajectory from h_{ini} is a hybrid trajectory halting in finite time, then the algorithm stops after a finite number of transitions. Secondly, if the hybrid trajectory is a chaotic hybrid trajectory, then Property 5 will be satisfied after a finite number of transitions, and once it is satisfied, the algorithm stops. Thirdly, if the hybrid trajectory is attracted by a cycle of discrete domains, then there are three cases: 1. The hybrid trajectory reaches R_{target} in finite time; 2. The hybrid trajectory reaches R_{target} after an infinite number of transitions;

3. The hybrid trajectory does not reach R_{target}. We assume here that Property 5 is not satisfied before the hybrid trajectory reaching the attractive cycle of discrete domains. For case 1, the algorithm stops in finite time, as the hybrid trajectory will eventually reach R_{target}. For case 2, the function *Stop_condition* returns "Reached" in finite time. For case 3, the function *Stop_condition* returns "Yes" in finite time. We need to mention that, since Property 5 is a necessary condition for the hybrid trajectory from h_{ini} to be a chaotic hybrid trajectory, a priori, it could be true even if the hybrid trajectory belongs to other classes; however, empirically, we never identified such case, which should be rare. Yet, an "unknown result" should be taken as a possible missed formal positive or negative case.

4.4 Application: Estimation of Basins of Attraction

In this section, we present a potential application of this reachability analysis method which is to estimate the basin of attraction of attractors. An attractor can be a stable fixed point, a stable limit cycle or a chaotic attractor. The basin of attraction of an attractor is the set of all hybrid states such that hybrid trajectories from these hybrid states converge to or reach this attractor. Here, estimating the basin of attraction means estimating the percentage of hybrid states which belong to the basin of attraction in a given region. A concrete example of application is presented in follows.

In this example, we consider a pre-existing 5-dimensional HGRN of cell cycle from [10, 13]. This HGRN has a stable limit cycle and a stable fixed point. We apply our reachability analysis method on this HGRN to estimate, for each discrete state d_s, the percentage of hybrid states which belong to the basin of attraction of this limit cycle, noted by $P_{LC}(d_s)$, and the percentage of hybrid states which belong to the basin of attraction of this stable fixed point, $P_{FP}(d_s)$.

To estimate these, we define two regions: R^{LC}_{target} and R^{FP}_{target}, such that for any hybrid trajectory, whether it reaches or converges to the stable limit cycle or the stable fixed point is equivalent to whether it reaches R^{LC}_{target} or R^{FP}_{target} respectively. These regions are found manually and defined as:

$R^{LC}_{target} = \{(\pi, (0,0,0,0,1)) \mid \pi^1 \in [0.82, 0.84], \pi^{2,3,4} \in [0, 0.01], \pi^5 \in [0.99, 1]\}$

$R^{FP}_{target} = \{(\pi, (2,1,0,1,0)) \mid \pi^{1,2,3,4,5} \in [0,1]\}$.

For R^{LC}_{target}, it can be easily verified that it contains hybrid states from the stable limit cycle and we can also prove that any hybrid trajectory which reaches this region must

reach this stable limit cycle. R^{FP}_{target} covers the whole discrete state which contains the stable fixed point, so reaching the stable fixed point is equivalent to reaching R^{FP}_{target}.

Then, in each discrete state d_s, we randomly generate n_s hybrid states by uniform distribution and we check whether these hybrid states reach R^{LC}_{target} or R^{FP}_{target} by using our reachability analysis method. We note $n_{LC}(d_s)$ the number of generated hybrid states that reach R^{LC}_{target} and we note $n_{FP}(d_s)$ the number of generated hybrid states that reach R^{FP}_{target}. $P_{LC}(d_s)$ and $P_{FP}(d_s)$ can be estimated by $P_{LC}(d_s) = \frac{n_{LC}}{n_s}$ and $P_{FP}(d_s) = \frac{n_{FP}}{n_s}$.

In our experimentation, we choose $n_s = 20$. A higher value of n_s can increase the precision of estimation, but it also takes more time. This setting takes nearly 30 minutes on a standard laptop computer [1]. The estimation results of P_{LC} and P_{FP} are illustrated in the transition graph of discrete states in Fig 4.6. For each node in Fig 4.6, the two floating point numbers represent estimated P_{LC} and P_{FP}. For example, for discrete state 20000, $P_{LC}(20000) = 0.75$ and $P_{FP}(20000) = 0.25$. The cycle of discrete states which contains the stable limit cycle is illustrated by the red arrows and the discrete state which contains the stable fixed point is illustrated by a red node. These results could provide useful information if we want to control a gene regulatory network such that the hybrid trajectory moves from one attractor to another.

4.5 Summary

In this chapter, we propose a reachability analysis method for HGRNs. In the first part of this work, we classify hybrid trajectories of HGRNs into different classes: Hybrid trajectories halting in finite time, hybrid trajectories attracted by cycles of discrete domains and chaotic hybrid trajectories, and provide some theoretical results about these hybrid trajectories regarding the reachability problem. Then, based on these theoretical results, we provide the first reachability analysis algorithm for HGRNs. Based on a first implementation, we apply this method on a 5-dimensional HGRN of cell cycle to estimate the basin of attraction of a stable limit cycle and a stable fixed point.

Since we use Property 5 (a necessary condition for chaotic hybrid trajectories) on line 14 in Algorithm 1, the algorithm might return inconclusive results ("unknown result") even in the cases that are decidable (non-chaotic hybrid trajectories). In fact, among HGRNs of gene regulatory networks, the cases that satisfy Property 5 are likely to be

1. Computations were performed on a standard laptop computer, with an Intel Core I7-8550U 1.80GHz processor and 16.0GB RAM.

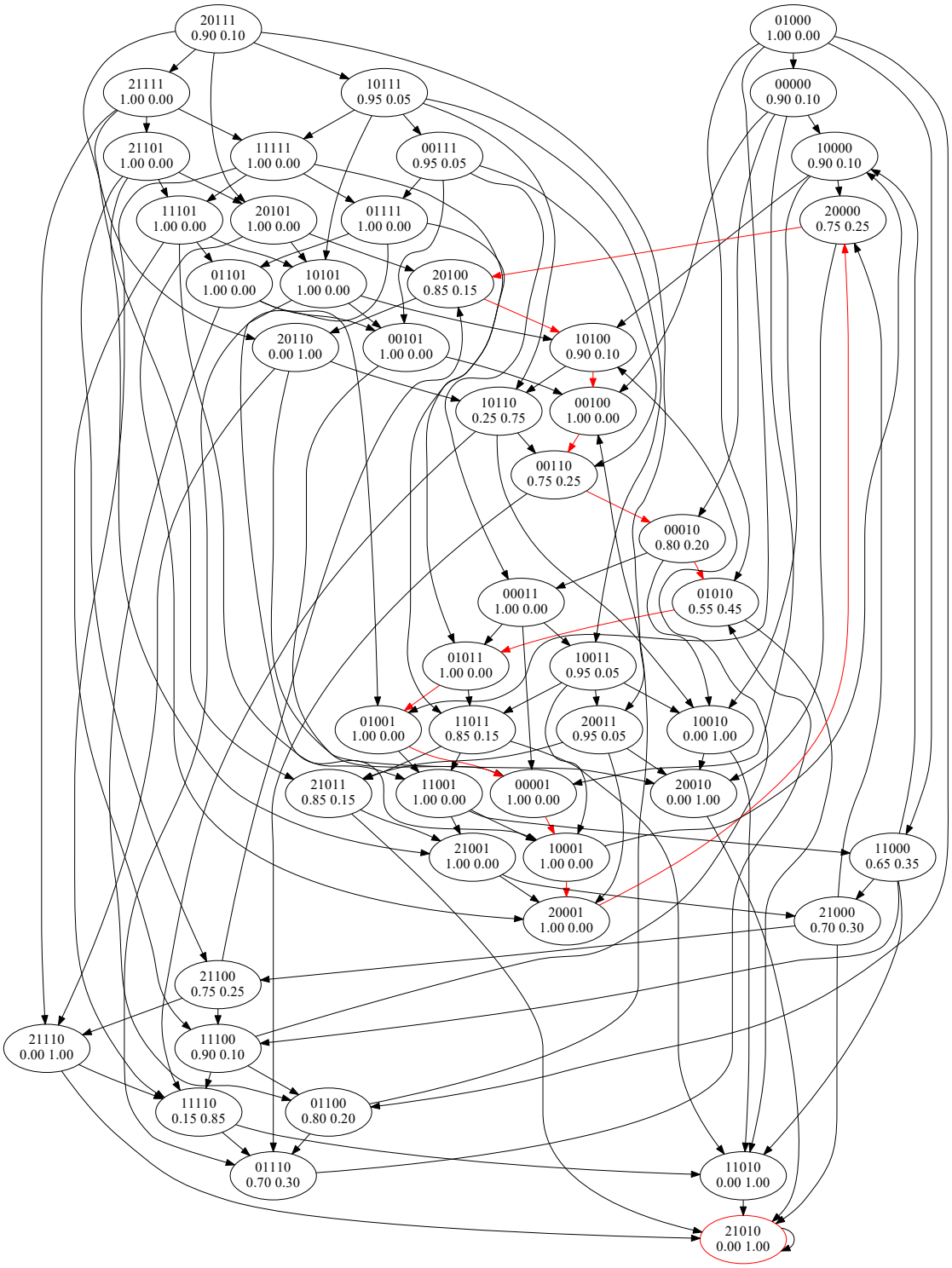


Figure 4.6 – Estimation results of P_{LC} and P_{FP} and illustration of the transition graph of discrete states

very rare: There is no identified HGRN of gene regulatory network with either chaos or non-chaotic hybrid trajectory that satisfies Property 5. The example with chaos in this chapter is not a biological model but comes from physics. For now, this algorithm can be considered sufficient for checking reachability in practice. However, it could be interesting to further investigate the decidability of chaotic hybrid trajectories and the condition for a hybrid trajectory to be chaotic.

This chapter and the previous one focus on the general analysis methods of HGRNs. These methods are very adapted to the study of oscillations in some specific networks. The following chapter focuses on this topic.

Chapter 5

Condition for Sustained Oscillation in Canonical Repressilator

The better control of synthetic oscillations in repressilator is an important question in synthetic biology. Most previous works use ordinary differential equations to study repressilators. In this chapter, we explore the possibility to use HGRN to study oscillations in the canonical repressialtor and aim to find conditions for the existence of sustained oscillations described by separable constraints on parameters. The methods in this chapter rely on the work in Chapter 3. Two major results are the following: 1) we develop, by using the Poincaré map, a sufficient and necessary condition for the existence of sustained oscillations (Section 5.4); 2) based on this condition, we give a method using the range enclosure property of Bernstein coefficients to compute compatible separable constraints (Section 5.5). The work of this chapter follows a 3-week research stay at the I3S laboratory (Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis) in Nice in June 2022 and has been presented at conference BIOINFORMATICS 2023 as regular paper [24].

5.1 Introduction of Canonical Repressilator

Repressilators represent gene regulatory networks consisting of at least one feedback loop, in which the expression of each gene inhibits the expression of the next gene. In this chapter, we focus on a specific repressilator called *canonical repressilator*, which is a network with three genes having a unique feedback loop with only inhibitions and not having any other regulation. The influence graph of the canonical repressilator is illustrated in Fig 5.1.

The canonical repressilator is widely studied in synthetic biology because it could

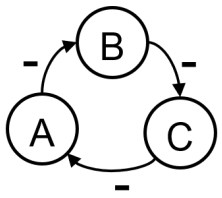


Figure 5.1 – The influence graph of the canonical repressilator.

generate synthetic sustained oscillations, which could be the treatment of diseases related to circadian rhythms, for instance, by allowing drug delivery at a particular pace. The first biological implementation of the canonical repressilator with sustained oscillations is realized in [57] by using three natural repressor proteins, the TetR, LacI and CI repressors and the stability of these oscillations is improved later in other works [58].

In fact, how to biologically implement a canonical repressilator with sustained oscillations is still an open question, particularly in eukaryotic cells, therefore exploring new models to search for conditions for sustained oscillations is of high interest.

Previous works about mathematical analysis of oscillations in the canonical repressilator are mainly based on differential equations. Many models of the canonical repressilator with three components are developed from a differential equation model using 6 variables [57], with 3 variables for repressor-protein concentrations and 3 variables for corresponding mRNA concentrations. This 6-variable model can also be reduced to a 3-variable model with only repressor-protein variables under certain assumptions [59]. These models and their variations are extensively studied in the literature [60–65]. One major limit about differential equations is that some dynamical properties are hard to analyze. So in this chapter, we use HGRNs to study oscillations in the canonical repressilator, as the dynamical properties of HGRNs are easier to analyse.

5.2 A HGRN of Canonical Repressilator

In this chapter, we assume that each gene has one threshold when it influences another gene. Although, *a priori*, one gene can have multiple thresholds for another gene, here we only consider the simplest case. Based on this assumption, since each gene only influences one other gene in the canonical repressilator, it has only two discrete levels separated by only one threshold.

The parameters (celerities) of this HGRN of the canonical repressilator are shown symbolically in Table 5.1. Each parameter in Table 5.1 is strictly positive and is denoted by C_{xyizj}, which represents the absolute value of the celerity of variable x when the discrete level of variable y is i and the discrete level of variable z is j. Consider the influence of A on B: When the discrete level of A is 1, meaning that the expression of A is above the threshold to inhibit B, then the time derivative of B is always negative, no matter the discrete level of B (0 or 1) which corresponds to the negative values $-C_{ba1b0}$ and $-C_{ba1b1}$. On the other hand, when the expression of A is below the threshold to inhibit B, the time derivative of B is always positive, corresponding to parameters C_{ba0b0} and C_{ba0b1}. From the parameters in this table, we can also see that the number of different parameters (12) is smaller than the multiplication of the number of dimensions by the number of discrete states (24), because some discrete states have celerities in common (same regulation on some variables).

In addition to the threshold which separates the discrete levels 0 and 1, each gene also has a maximal value and a minimal value. For example, when A inhibits B (see Figure 5.1), B will continue to decrease until it reaches the minimal value (most of the time this minimal value is 0) which is related to the lower boundary in the second dimension (the dimension of gene B) of discrete state $10*$, where $*$ can be 0 or 1. Similarly, when A does not inhibit B, B will continue to increase until it reaches the upper boundary in the second dimension of $01*$.

Table 5.1 – Parameters of the HGRN of the canonical repressilator.

A	B	C	C_A	C_B	C_C
0	0	0	C_{ac0a0}	C_{ba0b0}	C_{cb0c0}
0	0	1	$-C_{ac1a0}$	C_{ba0b0}	C_{cb0c1}
0	1	0	C_{ac0a0}	C_{ba0b1}	$-C_{cb1c0}$
0	1	1	$-C_{ac1a0}$	C_{ba0b1}	$-C_{cb1c1}$
1	0	0	C_{ac0a1}	$-C_{ba1b0}$	C_{cb0c0}
1	0	1	$-C_{ac1a1}$	$-C_{ba1b0}$	C_{cb0c1}
1	1	0	C_{ac0a1}	$-C_{ba1b1}$	$-C_{cb1c0}$
1	1	1	$-C_{ac1a1}$	$-C_{ba1b1}$	$-C_{cb1c1}$

Figure 5.2 gives two simulations with two different choices of parameters. The simulation on the left represents a sustained oscillation while the simulation on the right represents a damped oscillation. In the simulation on the left, gene C continues to increase from $t = 0$ until it reaches the maximal value, which also means that the hybrid

trajectory reaches an attractive boundary, then the hybrid trajectory will slide along this boundary (called the sliding mode) so that the value of gene C stays unchanged for some time.

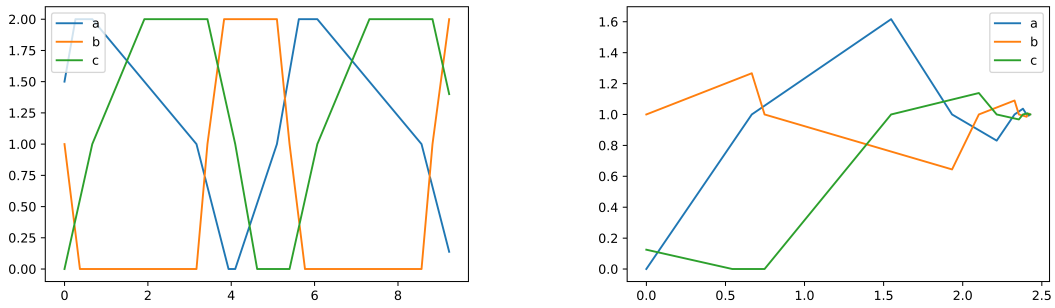


Figure 5.2 – Simulations of the HGRN of the canonical repressilator with two different choices of parameters (Abscissa represents time and ordinate represents the sum of the fractional part and the discrete state of each gene). Parameters of the model on the left: $C_{ac0a0} = 1$, $C_{ac0a1} = 1.9$, $C_{ac1a0} = 1.3$, $C_{ac1a1} = 0.4$, $C_{ba0b0} = 3.8$, $C_{ba0b1} = 2.5$, $C_{ba1b0} = 2.7$, $C_{ba1b1} = 3.3$, $C_{cb0c0} = 1.5$, $C_{cb0c1} = 0.8$, $C_{cb1c0} = 1.9$, $C_{cb1c1} = 1.5$. Parameters of the model on the right: $C_{ac0a0} = 1.5$, $C_{ac0a1} = 0.7$, $C_{ac1a0} = 0.6$, $C_{ac1a1} = 1.6$, $C_{ba0b0} = 2.1$, $C_{ba0b1} = 0.4$, $C_{ba1b0} = 0.3$, $C_{ba1b1} = 3.3$, $C_{cb0c0} = 1.25$, $C_{cb0c1} = 0.25$, $C_{cb1c0} = 0.23$, $C_{cb1c1} = 1.23$.

5.3 Qualitative Behaviors in this HGRN of Canonical Repressialtor

In this section, we discuss different qualitative properties of this HGRN. To analyze dynamical properties of a HGRN, we need to firstly analyze the transition graph of discrete states, which is determined by the signs of celerities. The signs of celerities of this HGRN of the repressilator can be found in Table 5.1, based on which the transition graph of discrete states can be constructed using the classical discrete asycnhronous semantics, see Figure 5.3.

From the transition graph, we can see that there is a unique cycle of discrete states, which is $001 \to 011 \to 010 \to 110 \to 100 \to 101 \to 001$. This cycle is a global attractor of discrete states (the unique terminal strongly connected component), which means that any hybrid trajectory in this HGRN will finally enter this cycle.

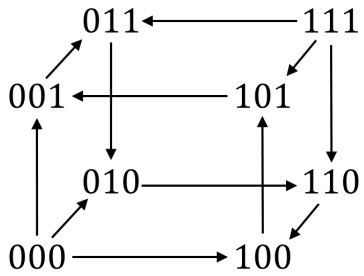


Figure 5.3 – Transition graph of discrete states of the HGRN of the canonical repressilator.

Following the same sequence of discrete states, there is a special hybrid trajectory: $((1,1,0),(0,0,1)) \rightarrow ((1,0,0),(0,1,1)) \rightarrow ((1,0,1),(0,1,0)) \rightarrow ((0,0,1),(1,1,0)) \rightarrow ((0,1,1),(1,0,0)) \rightarrow ((0,1,0),(1,0,1)) \rightarrow ((1,1,0),(0,0,1))$. It is illustrated in Figure 5.4 by green arrows. This hybrid trajectory contains 6 different hybrid states (the last and the first one are identical). From each hybrid state in this hybrid trajectory, there is an instant transition (transition which crosses boundaries between discrete states and takes no time), which reaches the next hybrid state. This hybrid trajectory represents also a closed hybrid trajectory, meaning that starting from each of these 6 hybrid states, the hybrid trajectory will return to the initial hybrid state. When representing hybrid trajectories, we often use an embedding of hybrid states in $\mathbb{R}^N$: A hybrid state (π, d_s) is represented in $\mathbb{R}^N$ by summing its discrete and fractional parts: $\pi + d_s$. We say that the hybrid state (π, d_s) and the point $\pi + d_s \in \mathbb{R}^N$ are related. When doing so, the six previous hybrid states are embedded in the same point H_0 in $\mathbb{R}^3$: $H_0 = (1.0, 1.0, 1.0)$. H_0 is called a *characteristic state* of this HGRN. The characteristic state is formally defined as follows:

Definition 14 (Characteristic state) *A characteristic state of a HGRN is a state H in Euclidean space such that: For any hybrid state h_0 related to H, all hybrid trajectories from h_0 will never reach a hybrid state which is not related to H, and there exist oscillations in any small neighborhood of H.*

In this chapter, we say that a hybrid trajectory of a HGRN is an oscillation if it is related to an oscillation in Euclidean space. Likewise, the nature of an oscillation of a HGRN (damped or sustained) and the relation between an oscillation and a characteristic state (converging to the characteristic state, moving away from it, etc.) are determined by the related oscillation in Euclidean space. A neighborhood of a characteristic state H is defined as a set $\mathcal{N}(H) = \{(\pi, d_s) \in E_h \mid \|\pi + d_s - H\| < r\}$ where $r \in \mathbb{R}$ is the radius of the neighborhood.

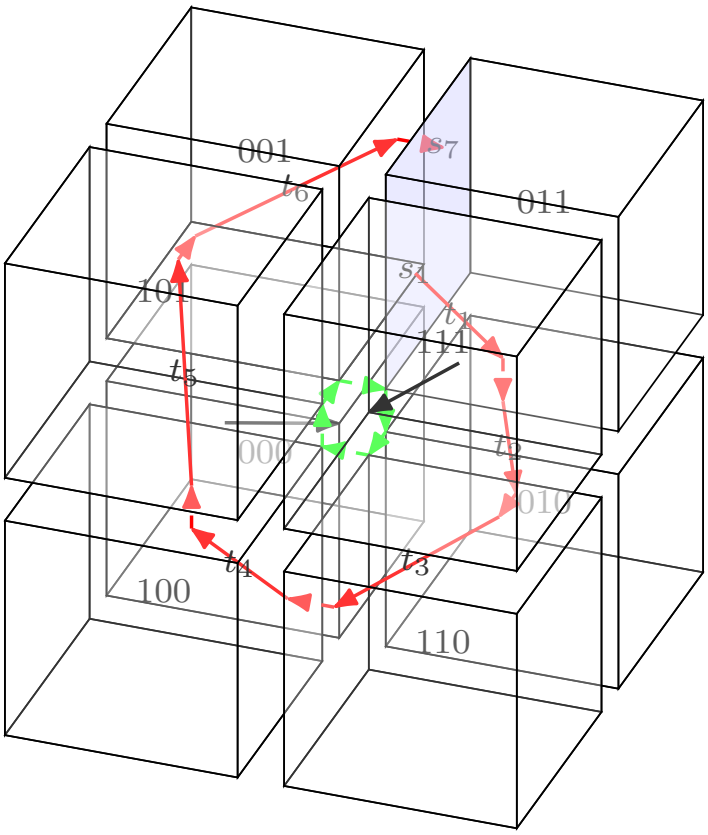


Figure 5.4 – Illustration of the closed hybrid trajectory with only instant transitions (green arrows), two special hybrid trajectories which can reach directly the characteristic state (black arrows), and a hybrid trajectory without sliding mode (red arrows) in this HGRN of the canonical repressilator.

In Euclidean space, a characteristic state is a fixed point, because from any hybrid state h related to a characteristic state, all hybrid trajectories can only reach hybrid states related to this characteristic state.

We can easily prove that H_0 is a characteristic state: Apart from the 6 hybrid states on this closed hybrid trajectory, there are two other hybrid states which are related to H_0: $((1,1,1),(0,0,0))$ and $((0,0,0),(1,1,1))$ From each of these two other hybrid states, all hybrid trajectories reach directly the closed hybrid trajectory, and once they do, they can never leave it. Finally, we can find oscillations which follow the unique cycle of discrete states in any neighborhood of H_0. In this HGRN, there is only one characteristic state. This can be proved by verifying all "corners" of discrete states.

All hybrid trajectories in this HGRN will oscillate in this unique cycle of discrete

states, except some special hybrid trajectories from discrete state 000 or 111 which can reach directly a hybrid state related to the characteristic state, see for example the black arrows in Figure 5.4. These oscillations can have several different dynamical properties.

To better illustrate the possible dynamical properties in this repressilator, different qualitative behaviors of a HGRN of negative feedback loop in 2 dimensions are illustrated in Figure 5.5, where black arrows represent celerities in each discrete state and red arrows represent some hybrid trajectories. The three figures are obtained by choosing three different parameterisations, which represent the three qualitative behaviors of this HGRN. The closed hybrid trajectory with hybrid states P, M, N, Q contains only instant transitions. The state in Euclidean space that is related to states P, M, N, Q is a characteristic state, and is the only characteristic state in this HGRN. On the left, the characteristic state is stable as all hybrid trajectories tend to converge to it. In this case, there is no sustained oscillation. On the right, the characteristic state is unstable as all hybrid trajectories from a small neighborhood of the characteristic state will move away from it and will finally reach a stable limit cycle containing at least one sliding mode. Between these two cases, in the middle, there is a third possibility in which all hybrid trajectories circle around the characteristic state without getting closer or moving away, which we call *parallel cycles* and can be considered as a special case of sustained oscillations.

The HGRN of the repressilator in 3 dimensions (Figure 5.1) is more complicated than the example HGRN of negative feedback loop in 2 dimensions (Figure 5.5), but we think that it also has three similar qualitative behaviors, since each discrete state in the only cycle of discrete states has only one successor (Figure 5.3). However, we have no proof for the non-existence of other possibilities yet, for example chaos or the co-existence of sustained oscillations and a stable fixed point. So in this chapter, we make Hypothesis 1.

Hypothesis 1 *In this HGRN of canonical repressilator (see Figure 5.1 and Table 5.1), either the characteristic state is stable and all oscillations are damped, or the characteristic state is unstable and all oscillations are sustained.*

In this HGRN of the canonical repressilator, the characteristic state is said *stable* if we can find a small neighborhood around the characteristic state such that all oscillations which begin from this neighborhood converge to the characteristic state, and the characteristic state is said *unstable* if there is no damped oscillation which converges to it.

Now, based on Hypothesis 1, we can use the stability of the characteristic state to determine the existence of sustained oscillations in this HGRN of the canonical repressilator:

All oscillations are sustained if and only if the characteristic state is unstable. This is the main idea used in this chapter to find conditions for the existence of sustained oscillations. A similar idea was used in other works with differential models, see for example [65–67].

The problem now is how to analyze the stability of the characteristic state. To do so, we apply a method based on the Poincaré map.

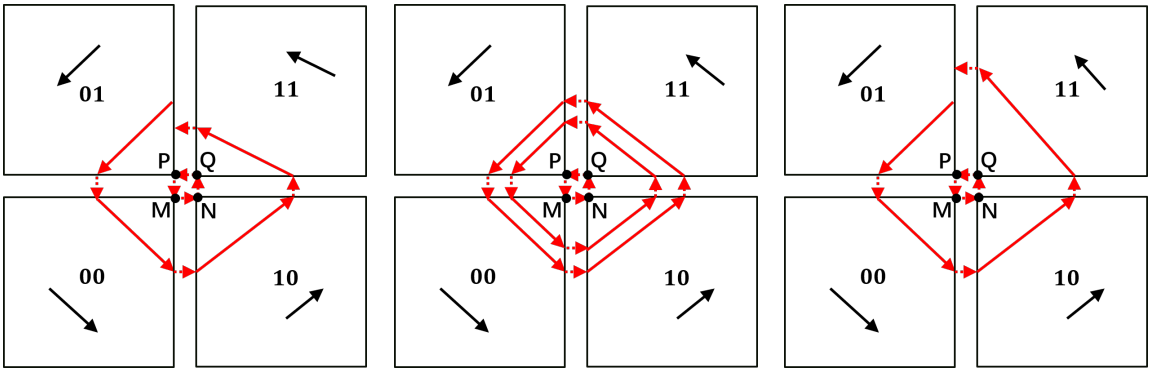


Figure 5.5 – Illustration of different qualitative behaviors in a HGRN of negative feedback loop in 2 dimensions. The three subfigures represent three different choices of parameters.

5.4 A Sufficient and Necessary Condition for Sustained Oscillation

In this section, firstly we present in details how we compute symbolically a Poincaré map that determines the stability of the characteristic state and then we develop a sufficient and necessary condition for sustained oscillation by symbolic eigenanalysis of this Poincaré map. The idea of the method to compute the Poincaré map in this chapter is same as the computation of the transition matrix in Chapter 3.

5.4.1 Symbolic Computation of Poincaré Map

In order to analyze the stability of the characteristic state, we only need to consider hybrid trajectories without sliding mode around this state. Indeed, the characteristic state is stable if we can find a small neighborhood around the characteristic state such that all oscillations which begin from this neighborhood converge to the characteristic state. Such a hybrid trajectory exists because otherwise, the celerities would prevent the characteristic state from existing. Without loss of generality, we choose the lower boundary

in the second dimension of discrete state 011 as Poincaré section; see the blue boundary in Figure 5.4. Now, we consider any hybrid trajectory τ which begins from a hybrid state $s_1 = ((x_1, 0, z_1), (0, 1, 1))$ on the Poincaré section and returns to the Poincaré section for the first time at $s_7 = ((x_7, 0, z_7), (0, 1, 1))$ without sliding mode; such a hybrid trajectory is illustrated in red in Figure 5.4. Thus, the Poincaré map is an affine application describing the relation between $(x_1, 0, z_1)$ and $(x_7, 0, z_7)$. Since s_1 is on an input boundary of the discrete state 011, from s_1, τ will first (continuously) cross the discrete state 011 and reach a hybrid state $((x_2, y_2, 0), (0, 1, 1))$ on the lower boundary in the third dimension of 011 which is the output boundary of this discrete state towards 010. We name this output boundary e_1. Then, it crosses instantly e_1 and reaches an input boundary of 010 in hybrid state $((x_2, y_2, 1), (0, 1, 0))$. The duration of crossing in discrete state 011 is:

$$t_1 = \frac{0 - z_1}{-C_{cb1c1}} \tag{5.1}$$

It should be noted that, to ensure that this hybrid trajectory τ has no sliding mode in 011, we also need to ensure that the lower boundary in the first dimension and the upper boundary in the second dimension of 011 are not reached before e_1, which gives us two additional inequalities:

$$t_1 < \frac{0 - x_1}{-C_{ac1a0}} \tag{5.2}$$

$$t_1 < \frac{1 - 0}{C_{ba0b1}} \tag{5.3}$$

In fact, these inequalities can always be satisfied if τ is sufficiently close to the characteristic state, which is the case we consider here. Therefore, in the rest of this section, we do not consider these additional constraints.

Based on the duration t_1 and the fact that there is no sliding mode, we can get the relation between $(x_1, 0, z_1)$ and $(x_2, y_2, 1)$:

$$x_2 = x_1 - C_{ac1a0} \times t_1 \tag{5.4}$$

$$y_2 = 0 + C_{ba0b1} \times t_1 \tag{5.5}$$

Following the same process, we can get the duration of τ in each discrete state and

the relations between hybrid states from one input boundary to another input boundary:

$$t_2 = \frac{1 - x_2}{C_{ac0a0}} \qquad t_3 = \frac{0 - y_3}{-C_{ba1b1}} \qquad t_4 = \frac{1 - z_4}{C_{cb0c0}} \tag{5.6}$$

$$t_5 = \frac{0 - x_5}{-C_{ac1a1}} \qquad t_6 = \frac{1 - y_6}{C_{ba0b0}} \tag{5.7}$$

$$y_3 = y_2 + C_{ba0b1} \times t_2 \qquad z_3 = 1 - C_{cb1c0} \times t_2 \tag{5.8}$$

$$x_4 = 0 + C_{ac0a1} \times t_3 \qquad z_4 = z_3 - C_{cb1c0} \times t_3 \tag{5.9}$$

$$x_5 = x_4 + C_{ac0a1} \times t_4 \qquad y_5 = 1 - C_{ba1b0} \times t_4 \tag{5.10}$$

$$y_6 = y_5 - C_{ba1b0} \times t_5 \qquad z_6 = 0 + C_{cb0c1} \times t_5 \tag{5.11}$$

$$x_7 = 1 - C_{ac1a0} \times t_6 \qquad z_7 = z_6 + C_{cb0c1} \times t_6 \tag{5.12}$$

where t_2, t_3, t_4, t_5, t_6 are the durations of τ in discrete states 010, 110, 100, 101, 001 respectively, and $(0, y_3, z_3), (x_4, 1, z_4), (x_5, y_5, 0), (1, y_6, z_6), (x_7, 0, z_7)$ are the fractional parts of the hybrid states when τ first reaches 110, 100, 101, 001, 011 respectively.

Based on the above equations, the Poincaré map can be calculated to describe the relation between $(x_1, 0, z_1)$ and $(x_7, 0, z_7)$ as follows. One dimension is missing in the matrix below; indeed, this dimension is useless in the computation of the stability, for more details see Chapter 3 Section 3.3.2.

$$\begin{pmatrix} x_7 \\ z_7 \end{pmatrix} = \begin{bmatrix} b_1 & c_1 \\ b_2 & c_2 \end{bmatrix} \begin{pmatrix} x_1 \\ z_1 \end{pmatrix} + \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \tag{5.13}$$

In the above equation, a_1, a_2, b_1, b_2, c_1, c_2 are nonlinear combinations of the celerity values. Their expressions are given in the follows. We can easily derive that b_1 and c_2 are strictly positive, while b_2 and c_1 are strictly negative.

$$b_1 = \frac{B_1}{D_0} \quad b_2 = \frac{B_2}{D_0} \quad c_1 = \frac{C_1}{D_0} \quad c_2 = \frac{C_2}{D_0} \tag{5.14}$$

$$\begin{aligned} B_1 = {} & C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb0c0}C_{cb1c1} + C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb1c0}C_{cb1c1} \\ & + C_{ac0a1}C_{ac1a0}C_{ba1b0}C_{ba1b1}C_{cb1c0}C_{cb1c1} + C_{ac1a0}C_{ac1a1}C_{ba0b1}C_{ba1b0}C_{cb1c0}C_{cb1c1} \\ & + C_{ac1a0}C_{ac1a1}C_{ba1b0}C_{ba1b1}C_{cb1c0}C_{cb1c1} \end{aligned} \tag{5.15}$$

$$\begin{aligned}B_2 = & - C_{ac0a1}C_{ba0b0}C_{ba0b1}C_{cb0c0}C_{cb0c1}C_{cb1c1} - C_{ac0a1}C_{ba0b0}C_{ba0b1}C_{cb0c1}C_{cb1c0}C_{cb1c1} \\ & - C_{ac0a1}C_{ba0b0}C_{ba1b1}C_{cb0c1}C_{cb1c0}C_{cb1c1} - C_{ac0a1}C_{ba0b1}C_{ba1b0}C_{cb0c0}C_{cb0c1}C_{cb1c1} \\ & - C_{ac0a1}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0}C_{cb1c1} - C_{ac0a1}C_{ba1b0}C_{ba1b1}C_{cb0c1}C_{cb1c0}C_{cb1c1} \\ & - C_{ac1a1}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0}C_{cb1c1} - C_{ac1a1}C_{ba1b0}C_{ba1b1}C_{cb0c1}C_{cb1c0}C_{cb1c1}\end{aligned} \tag{5.16}$$

$$\begin{aligned}C_1 = & - C_{ac0a0}C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb0c0} - C_{ac0a0}C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb1c0} \\ & - C_{ac0a0}C_{ac1a0}C_{ac1a1}C_{ba0b1}C_{ba1b0}C_{cb1c0} - C_{ac0a1}C_{ac1a0}^2C_{ba0b1}C_{ba1b0}C_{cb0c0} \\ & - C_{ac0a1}C_{ac1a0}^2C_{ba0b1}C_{ba1b0}C_{cb1c0} - C_{ac0a1}C_{ac1a0}^2C_{ba1b0}C_{ba1b1}C_{cb1c0} \\ & - C_{ac1a0}^2C_{ac1a1}C_{ba0b1}C_{ba1b0}C_{cb1c0} - C_{ac1a0}^2C_{ac1a1}C_{ba1b0}C_{ba1b1}C_{cb1c0}\end{aligned} \tag{5.17}$$

$$\begin{aligned}C_2 = & C_{ac0a0}C_{ac0a1}C_{ba0b0}C_{ba0b1}C_{cb0c0}C_{cb0c1} + C_{ac0a0}C_{ac0a1}C_{ba0b0}C_{ba0b1}C_{cb0c1}C_{cb1c0} \\ & + C_{ac0a0}C_{ac0a1}C_{ba0b1}C_{ba1b0}C_{cb0c0}C_{cb0c1} + C_{ac0a0}C_{ac0a1}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0} \\ & + C_{ac0a0}C_{ac1a1}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0} + C_{ac0a1}C_{ac1a0}C_{ba0b0}C_{ba0b1}C_{cb0c0}C_{cb0c1} \\ & + C_{ac0a1}C_{ac1a0}C_{ba0b0}C_{ba0b1}C_{cb0c1}C_{cb1c0} + C_{ac0a1}C_{ac1a0}C_{ba0b0}C_{ba1b1}C_{cb0c1}C_{cb1c0} \\ & + C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb0c0}C_{cb0c1} + C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0} \\ & + C_{ac0a1}C_{ac1a0}C_{ba1b0}C_{ba1b1}C_{cb0c1}C_{cb1c0} + C_{ac1a0}C_{ac1a1}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0} \\ & + C_{ac1a0}C_{ac1a1}C_{ba1b0}C_{ba1b1}C_{cb0c1}C_{cb1c0}\end{aligned} \tag{5.18}$$

$$D_0 = C_{ac0a0}C_{ac1a1}C_{ba0b0}C_{ba1b1}C_{cb0c0}C_{cb1c1} \tag{5.19}$$

5.4.2 Symbolic Eigenanalysis

The stability of the characteristic state depends on the two eigenvalues of $\begin{bmatrix} b_1 & c_1 \\ b_2 & c_2 \end{bmatrix}$, which are:

$$\lambda_1 = \frac{b_1 + c_2 + \sqrt{(b_1 - c_2)^2 + 4c_1b_2}}{2} \tag{5.20}$$

$$\lambda_2 = \frac{b_1 + c_2 - \sqrt{(b_1 - c_2)^2 + 4c_1b_2}}{2} \tag{5.21}$$

Property 6 *These two eigenvalues are real and strictly positive.*

Proof: From the expressions of b_1, b_2, c_1, c_2, we know that $b_1 > 0$, $b_2 < 0$, $c_1 < 0$, $c_2 > 0$, so we have $(b_1 - c_2)^2 + 4c_1b_2 > 0$, therefore these two eigenvalues are real. Moreover, the

expression of the product of these two eigenvalues is:

$$\lambda_1 \times \lambda_2 = b_1c_2 - c_1b_2 = \frac{D_{\lambda_1\times\lambda_2}}{D_0} \tag{5.22}$$

where

$$D_{\lambda_1\times\lambda_2} = C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0} \tag{5.23}$$

$$D_0 = C_{ac0a0}C_{ac1a1}C_{ba0b0}C_{ba1b1}C_{cb0c0}C_{cb1c1} \tag{5.24}$$

Therefore, $\lambda_1 \times \lambda_2 > 0$. Since λ_1 is strictly positive, so is λ_2. □

Suppose that two eigenvectors which are related to λ_1 and λ_2 respectively are $v_1 = (v_1^1, v_1^2)$ and $v_2 = (v_2^1, v_2^2)$. We have the following property on v_1 and v_2.

Property 7 *$v_1^1 \times v_1^2 < 0$ and $v_2^1 \times v_2^2 > 0$.*

Proof: Since v_1 is a eigenvector related to λ_1, we have:

$$\lambda_1 \begin{pmatrix} v_1^1 \\ v_1^2 \end{pmatrix} = \begin{bmatrix} b_1 & c_1 \\ b_2 & c_2 \end{bmatrix} \begin{pmatrix} v_1^1 \\ v_1^2 \end{pmatrix} \tag{5.25}$$

From which, we have:

$$v_1^2 = \frac{\lambda_1 - b_1}{c_1} v_1^1 \tag{5.26}$$

By developing the expression $\frac{\lambda_1 - b_1}{c_1}$, we have:

$$\frac{\lambda_1 - b_1}{c_1} = \frac{c_2 - b_1 + \sqrt{(c_2 - b_1)^2 + 4c_1b_2}}{2c_1} \tag{5.27}$$

We can see that whether $c_2 - b_1$ is positive or negative, $\lambda_1 - b_1$ is always strictly positive because $|c_2 - b_1| < \sqrt{(c_2 - b_1)^2 + 4c_1b_2}$. Since c_1 is strictly negative, we have $v_1^1 \times v_1^2 < 0$.

Similarly for v_2, we can have:

$$v_2^2 = \frac{\lambda_2 - b_1}{c_1} v_2^1 \tag{5.28}$$

and

$$\frac{\lambda_2 - b_1}{c_1} = \frac{c_2 - b_1 - \sqrt{(c_2 - b_1)^2 + 4c_1b_2}}{2c_1} \tag{5.29}$$

So, similarly, whether $c_2 - b_1$ is positive or negative, $\lambda_2 - b_1$ is always strictly negative. And since c_1 is strictly negative, we have $v_2^1 \times v_2^2 > 0$. □

Based on these properties, we develop the following theorem to verify the stability of the characteristic state.

Theorem 7 *The characteristic state is unstable if and only if* $\lambda_1 \geq 1$.

Proof: For any hybrid trajectory τ sufficiently close to the characteristic state but different from the characteristic state, which reaches the chosen Poincaré section at a hybrid state $s = ((x_1, 0, z_1), (0, 1, 1))$, and then reaches this Poincaré section again for the first time at another hybrid state $s' = ((x_2, 0, z_2), (0, 1, 1))$, such that there is no sliding mode between s and s', since the hybrid state $((1, 0, 0), (0, 1, 1))$ (which is related to the characteristic state) is a fixed point of the Poincaré map, we can have the following relation:

$$\begin{pmatrix} x_2' \\ z_2' \end{pmatrix} = \begin{bmatrix} b_1 & c_1 \\ b_2 & c_2 \end{bmatrix} \begin{pmatrix} x_1' \\ z_1' \end{pmatrix} \tag{5.30}$$

where (x_1', z_1') and (x_2', z_2') are the new coordinates of (x_1, z_1) and (x_2, z_2) by taking $(1, 0)$ as the new origin, which means that $x_i' = x_i - 1$, $z_i' = z_i - 0$, where $i \in \{1, 2\}$. In fact, this change of coordinates allows to remove the affine vector $\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$ in Equation 5.13.

Vectors (x_1', z_1') and (x_2', z_2') can be decomposed by:

$$\begin{pmatrix} x_1' \\ z_1' \end{pmatrix} = \alpha_1 v_1 + \beta_1 v_2 \tag{5.31}$$

$$\begin{pmatrix} x_2' \\ z_2' \end{pmatrix} = \alpha_2 v_1 + \beta_2 v_2 \tag{5.32}$$

where $\alpha_1, \beta_1, \alpha_2, \beta_2 \in \mathbb{R}$ and α_1 is not null as $x_1' < 0$, $z_1' > 0$, $v_1^1 \times v_1^2 < 0$ and $v_2^1 \times v_2^2 > 0$.

By multiplying Equation 5.31 on the left by $\begin{bmatrix} b_1 & c_1 \\ b_2 & c_2 \end{bmatrix}$ and from Equation (5.30), we can derive that:

$$\alpha_2 = \lambda_1 \alpha_1 \tag{5.33}$$

$$\beta_2 = \lambda_2 \beta_1 \tag{5.34}$$

Consider the case where $\lambda_1 \geq 1$. Since $|\alpha_1|$ is strictly positive, then we have $|\alpha_2| \geq |\alpha_1|$, which means that s' is not closer to the characteristic state than s in the direction of v_1.

Therefore, any sequence of the intersection points of τ with this Poincaré section will never converge to the characteristic state, which indicates that the characteristic state is unstable. This proves that "the characteristic state is unstable" is a necessary condition for "$\lambda_1 \geq 1$".

Now consider the case where the characteristic state is unstable, meaning that there is no damped oscillation that converges to it. We suppose that $\lambda_1 < 1$. Since $\lambda_1 > \lambda_2 > 0$, we have $|\alpha_2| < |\alpha_1|$ and $|\beta_2| < |\beta_1|$, which means that if τ is sufficiently close to the characteristic state, then it converges to the characteristic state, which contradicts the hypothesis stating that the characteristic state is unstable. This proves that "the characteristic state is unstable" is a sufficient condition for "$\lambda_1 \geq 1$". □

Based on Hypothesis 1 and Theorem 7, the condition $\lambda_1 \geq 1$ is a sufficient and necessary condition for the existence of sustained oscillations in this HGRN of canonical repressilator.

Since our final objective is to provide practical information for the construction of synthetic networks, conditions like $\lambda_1 \geq 1$ might not be a good result, because the set of models under this constraint is not easy to figure. In the next section, a method to compute separable constraints based on the condition $\lambda_1 \geq 1$ is presented.

5.5 Computation of Sufficient Separable Constraints on Parameters

In this section, we propose a method to compute separable constraints on parameters based on the condition $\lambda_1 \geq 1$ which is developed in the previous section. In this chapter, "separable constraints" mean constraints with separable form: Each parameter is constrained by an interval, for example $C_{ac0a0} \in [\underline{C_{ac0a0}}, \overline{C_{ac0a0}}]$. The reason why we choose constraints of separable form is that they can be easily interpreted and used. In other words, these separable constraints represent a n-dimensional bounding box in the space of parameters (celerities). What we want to do is to find such a n-dimensional box so that any model in this bounding box satisfies the condition $\lambda_1 \geq 1$, which means that any model in this box has sustained oscillations.

Firstly, we present a simplification of the condition $\lambda_1 \geq 1$. Secondly, we introduce a method to verify if all models in a given bounding box satisfy this simplified condition. At last, using the method in the second part, we propose a search algorithm to find some separable constraints.

5.5.1 Condition Simplification

The condition:

$$\lambda_1 = \frac{b_1 + c_2 + \sqrt{(b_1 - c_2)^2 + 4c_1b_2}}{2} \geq 1 \tag{5.35}$$

can be reformulated as:

$$b_1 + c_2 - 2 \geq -\sqrt{(b_1 - c_2)^2 + 4c_1b_2} \tag{5.36}$$

which is equivalent to:

$$(b_1 + c_2 - 2 \geq 0) \vee ((b_1 + c_2 - 2 < 0) \wedge ((b_1 + c_2 - 2)^2 \leq (b_1 - c_2)^2 + 4c_1b_2)) \tag{5.37}$$

or:

$$(b_1 + c_2 - 2 \geq 0) \vee (b_1c_2 - c_1b_2 - b_1 - c_2 + 1 \leq 0) \tag{5.38}$$

This last condition is equivalent to $(P_1 \geq 0) \vee (P_2 \geq 0)$ where P_1 and P_2 are polynomials on parameters. The expressions of P_1 and P_2 are given as follows.

$$\begin{aligned}
P_1 = {} & C_{ac0a0}C_{ac0a1}C_{ba0b0}C_{ba0b1}C_{cb0c0}C_{cb0c1} + C_{ac0a0}C_{ac0a1}C_{ba0b0}C_{ba0b1}C_{cb0c1}C_{cb1c0} \\
& + C_{ac0a0}C_{ac0a1}C_{ba0b1}C_{ba1b0}C_{cb0c0}C_{cb0c1} + C_{ac0a0}C_{ac0a1}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0} \\
& - 2C_{ac0a0}C_{ac1a1}C_{ba0b0}C_{ba1b1}C_{cb0c0}C_{cb1c1} + C_{ac0a0}C_{ac1a1}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0} \\
& + C_{ac0a1}C_{ac1a0}C_{ba0b0}C_{ba0b1}C_{cb0c0}C_{cb0c1} + C_{ac0a1}C_{ac1a0}C_{ba0b0}C_{ba0b1}C_{cb0c1}C_{cb1c0} \\
& + C_{ac0a1}C_{ac1a0}C_{ba0b0}C_{ba1b1}C_{cb0c1}C_{cb1c0} + C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb0c0}C_{cb0c1} \\
& + C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb0c0}C_{cb1c1} + C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0} \\
& + C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb1c0}C_{cb1c1} + C_{ac0a1}C_{ac1a0}C_{ba1b0}C_{ba1b1}C_{cb0c1}C_{cb1c0} \\
& + C_{ac0a1}C_{ac1a0}C_{ba1b0}C_{ba1b1}C_{cb1c0}C_{cb1c1} + C_{ac1a0}C_{ac1a1}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0} \\
& + C_{ac1a0}C_{ac1a1}C_{ba0b1}C_{ba1b0}C_{cb1c0}C_{cb1c1} + C_{ac1a0}C_{ac1a1}C_{ba1b0}C_{ba1b1}C_{cb0c1}C_{cb1c0} \\
& + C_{ac1a0}C_{ac1a1}C_{ba1b0}C_{ba1b1}C_{cb1c0}C_{cb1c1}
\end{aligned} \tag{5.39}$$

$$
\begin{aligned}
P_2 = {} & C_{ac0a0}C_{ac0a1}C_{ba0b0}C_{ba0b1}C_{cb0c0}C_{cb0c1} + C_{ac0a0}C_{ac0a1}C_{ba0b0}C_{ba0b1}C_{cb0c1}C_{cb1c0} \\
& + C_{ac0a0}C_{ac0a1}C_{ba0b1}C_{ba1b0}C_{cb0c0}C_{cb0c1} + C_{ac0a0}C_{ac0a1}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0} \\
& - C_{ac0a0}C_{ac1a1}C_{ba0b0}C_{ba1b1}C_{cb0c0}C_{cb1c1} + C_{ac0a0}C_{ac1a1}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0} \\
& + C_{ac0a1}C_{ac1a0}C_{ba0b0}C_{ba0b1}C_{cb0c0}C_{cb0c1} + C_{ac0a1}C_{ac1a0}C_{ba0b0}C_{ba0b1}C_{cb0c1}C_{cb1c0} \\
& + C_{ac0a1}C_{ac1a0}C_{ba0b0}C_{ba1b1}C_{cb0c1}C_{cb1c0} + C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb0c0}C_{cb0c1} \\
& + C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb0c0}C_{cb1c1} + C_{ac0a1}C_{ac1a0}C_{ba0b1}C_{ba1b0}C_{cb1c0}C_{cb1c1} \\
& + C_{ac0a1}C_{ac1a0}C_{ba1b0}C_{ba1b1}C_{cb0c1}C_{cb1c0} + C_{ac0a1}C_{ac1a0}C_{ba1b0}C_{ba1b1}C_{cb1c0}C_{cb1c1} \\
& + C_{ac1a0}C_{ac1a1}C_{ba0b1}C_{ba1b0}C_{cb0c1}C_{cb1c0} + C_{ac1a0}C_{ac1a1}C_{ba0b1}C_{ba1b0}C_{cb1c0}C_{cb1c1} \\
& + C_{ac1a0}C_{ac1a1}C_{ba1b0}C_{ba1b1}C_{cb0c1}C_{cb1c0} + C_{ac1a0}C_{ac1a1}C_{ba1b0}C_{ba1b1}C_{cb1c0}C_{cb1c1}
\end{aligned} \tag{5.40}
$$

Condition $(P_1 \geq 0) \vee (P_2 \geq 0)$ seems preferable to $\lambda_1 \geq 1$ because it only contains polynomials. In fact, one can easily prove that solutions for $(P_1 \geq 0) \vee (P_2 \geq 0)$ exist. For example, by only considering $Cac0a0$ and $Cac0a1$, which are two parameters describing the derivative of gene A when gene C does not inhibit gene A, P_1 and P_2 can be expressed by:

$$P_1 = p_{11} \times C_{ac0a0} + p_{12} \times C_{ac0a1} + p_{13} \times C_{ac0a0} \times C_{ac0a1} + p_{14} \tag{5.41}$$

$$P_2 = p_{21} \times C_{ac0a0} + p_{22} \times C_{ac0a1} + p_{23} \times C_{ac0a0} \times C_{ac0a1} + p_{24} \tag{5.42}$$

where p_{ij} (with $i \in \{1, 2\}$, $j \in \{1, 2, 3, 4\}$) are expressions of parameters which do not include C_{ac0a0} and C_{ac0a1}. We can see that if C_{ac0a0} and C_{ac0a1} converge to 0 while other parameters remain unchanged, P_1 and P_2 converge to p_{14} and p_{24} respectively, which are both positive. This indicates that the solutions of $(P_1 \geq 0) \vee (P_2 \geq 0)$ exist and also implies a new control strategy for the existence of sustained oscillations: Controlling the derivatives of gene A when A is not inhibited by C such that these derivatives are sufficiently small, while keeping other parameters unchanged.

5.5.2 Satisfiability under Separable Constraints

In this subsection, we adapt the range enclosure property of Bernstein coefficients to verify if all models in a given bounding box satisfy the condition $(P_1 \geq 0) \vee (P_2 \geq 0)$. The Bernstein coefficients have been used in the literature to, for example, compute images for polynomial dynamical system [50, 68], or compute affine lower bound functions for polynomials [69], etc.

Before introducing the Bernstein coefficients, we firstly introduce the notion of multi-indice. A multi-indice is a vector of non-negative integers. Given two multi-indice $i = (i_1, i_2, ..., i_n)$ and $j = (j_1, j_2, ..., j_n)$, we write $i \leq j$ if $\forall k \in \{1, 2, ..., n\}$, $i_k \leq j_k$. We also write $\frac{i}{j}$ for $(\frac{i_1}{j_1}, \frac{i_2}{j_2}, ..., \frac{i_n}{j_n})$ and $\binom{j}{i}$ for $\binom{j_1}{i_1}\binom{j_2}{i_2}...\binom{j_n}{i_n}$ which is the multiplication of all binomial coefficient $\binom{j_k}{i_k}$, $k \in \{1, 2, ..., n\}$.

Using the multi-indice, a polynomial $f : \mathbb{R}^n \to \mathbb{R}$ can be represented as follows:

$$f(x) = \sum_{i \in I_d} a_i x^i \tag{5.43}$$

where $a_i \in \mathbb{R}$, i and d are multi-indices, I_d is the set of all multi-indices i such that $i \leq d$, and $x^i = x_1^{i_1} x_2^{i_2} ... x_n^{i_n}$ is the product of all $x_j^{i_j}$, where x_j is the j^{th} variable of polynomial f.

f can also be expressed by Bernstein expansion as follows:

$$f(x) = \sum_{i \in I_d} b_i \mathcal{B}_{d,i}(x) \tag{5.44}$$

where

$$\mathcal{B}_{d,i}(x) = \beta_{d_1,i_1}(x_1) ... \beta_{d_n,i_n}(x_n) \tag{5.45}$$

$$\beta_{d_k,i_k}(x_k) = \binom{d_k}{i_k} x_k^{i_k} (1 - x_k)^{d_k - i_k} \tag{5.46}$$

$$b_i = \sum_{j \leq i} \frac{\binom{i}{j}}{\binom{d}{j}} a_j \tag{5.47}$$

where d and i are multi-indices and $k \in \{1, 2, ..., n\}$. The values b_i, for $i \in I_d$ are called Bernstein coefficients.

One fundamental property of Bernstein coefficients for our approach is the range enclosure property, which can be derived from the convex hull property. The convex hull of a set S, noted $Conv(S)$, is the smallest convex set that contains S.

Lemma 1 (Convex hull property) $Conv\{(x, f(x)) \mid x \in B\} \subseteq Conv\{(i/d, b_i) \mid i \in I_d\}$.

Lemma 2 (Range enclosure property) $\min\{b_i \mid i \in I_d\} \leq f(x) \leq \max\{b_i \mid i \in I_d\}$,

$\forall x \in B$, *where* $B = [0,1]^n$ *is the unit box.*

The range enclosure property over-approximates the range of the image of f on B and can be used to verify if all models in a given bounding box satisfy the condition $(P_1 \geq 0) \vee (P_2 \geq 0)$. To do so, we need to firstly make a change of variables of polynomials P_1 and P_2, such that all variables are included in $[0,1]$. For example, the variable $C_{ac0a0} \in [\underline{C_{ac0a0}}, \overline{C_{ac0a0}}]$ is replaced by $C_{ac0a0} = (\overline{C_{ac0a0}} - \underline{C_{ac0a0}}) \times X_{ac0a0} + \underline{C_{ac0a0}}$, where $X_{ac0a0} \in [0,1]$. By doing so, we get two new polynomials P_1' and P_2'.

So now, to verify if $(P_1 \geq 0) \vee (P_2 \geq 0)$ is always true in a given bounding box, we only need to verify if $(P_1' \geq 0) \vee (P_2' \geq 0)$ is always true in the unit box. To do so, we compute the Bernstein coefficients $\{b_{1,i}\}$ and $\{b_{2,i}\}$ (where $i \in I_d$) of P_1' and P_2' respectively. A sufficient condition for the condition "$(P_1' \geq 0) \vee (P_2' \geq 0)$ is always true in the unit box" (condition1) is "$(\forall i \in I_d,\ b_{1,i} \geq 0) \vee (\forall i \in I_d,\ b_{2,i} \geq 0)$" (condition2), according to the range enclosure property. In fact, since the minimum value of the image of P_1' on the unit box is always larger or equal to the minimum value of $\{b_{1,i}\}$, "$\{b_{1,i}\}$ are not negative" ($\forall i \in I_d,\ b_{1,i} \geq 0$) indicates that "$P_1'$ is not negative on the unit box", and the same holds for P_2'. Since there is a finite number of Bernstein coefficients, condition2 can be verified. Therefore, in this chapter, condition2 is used to verify if all models in a given bounding box always have sustained oscillations.

5.5.3 Search of Separable Constraints

Based on the method introduced in the previous subsection, we propose a depth first algorithm to find some bounding boxes which satisfy condition2. This algorithm is illustrated in Figure 5.6. Initially, each parameter is included in an interval. In this implementation, without loss of generality, we assume that each parameter is included initially in $[0,1]$. Then, we verify if condition2 is satisfied for this bounding box using the method proposed in the previous subsection. If it is satisfied, then it is a bounding box such that all models in it have sustained oscillations. If condition2 is not satisfied, then there might be some models in this bounding box which do not have sustained oscillations, in this case the bounding box is split into two smaller bounding boxes (by splitting the largest interval into two) which have the same volume and the process is repeated on each of these two new bounding boxes. Each path in this algorithm will stop, either when a bounding box which satisfies condition2 is found or when the length of the largest interval is smaller than a certain threshold. In fact, similar ideas are widely used to find solution sets under

non-linear constraints [70, 71].

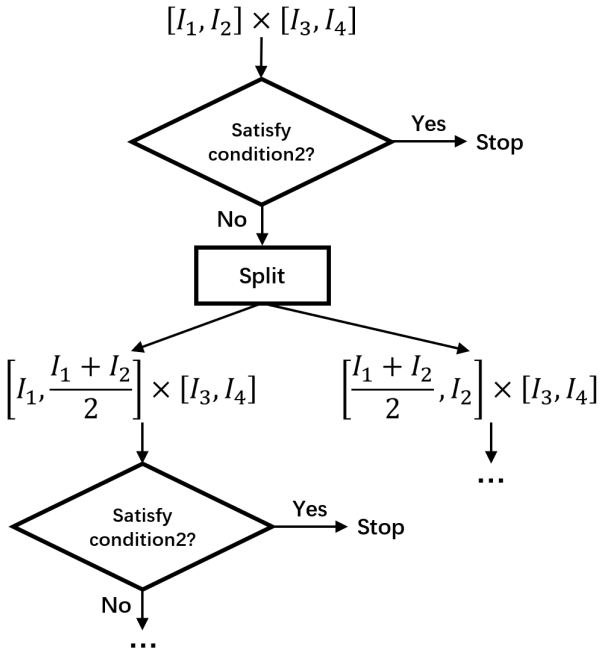


Figure 5.6 – Illustration of the algorithm to search for some bounding boxes.

Since this HGRN of the canonical repressilator has 12 parameters (see Table 5.1), and if we assume that the number of possible smallest intervals for each parameter are the same, noted m, then there are at most m^{12} smallest bounding boxes to check. Verifying all these boxes can be time consuming. In our implementation, we assume that the intervals of these three genes are identical, which means that we search for bounding boxes such that $\underline{C_{aciaj}} = \underline{C_{baibj}} = \underline{C_{cbicj}}$ and $\overline{C_{aciaj}} = \overline{C_{baibj}} = \overline{C_{cbicj}}$ for any $i, j \in \{0, 1\}$, where, for example, $[\underline{C_{ac0a0}}, \overline{C_{ac0a0}}]$ is the interval of the parameter C_{ac0a0}, so that we only need to consider 4 independent intervals when searching for bounding boxes. This assumption is only applied here to decrease the number of possible bounding boxes. A similar assumption about the symmetry between these three genes was also made in works based on differential equations, see for example [59]. We also assume that the minimal length of interval is greater or equal to 0.5. A value smaller than 0.5 could naturally be chosen, but this might exponentially increase the number of possible bounding boxes, which could also exponentially increase the execution time. With these assumptions, we obtain 5 bounding boxes which satisfy the condition2. In the results below, (y, x) represents (c, a), (a, b) and (b, c), where y inhibits x, for example C_{xy0x0} presents C_{ac0a0}, C_{ba0b0} and C_{cb0c0}.

Bounding box 1: $C_{xy0x0} \in [0, 0.5], C_{xy0x1} \in [0, 0.5], C_{xy1x0} \in [0.5, 1], C_{xy1x1} \in [0, 1]$
Bounding box 2: $C_{xy0x0} \in [0, 0.5], C_{xy0x1} \in [0.5, 1], C_{xy1x0} \in [0, 0.5], C_{xy1x1} \in [0, 0.5]$
Bounding box 3: $C_{xy0x0} \in [0, 0.5], C_{xy0x1} \in [0.5, 1], C_{xy1x0} \in [0.5, 1], C_{xy1x1} \in [0, 1]$
Bounding box 4: $C_{xy0x0} \in [0.5, 1], C_{xy0x1} \in [0.5, 1], C_{xy1x0} \in [0, 0.5], C_{xy1x1} \in [0, 0.5]$
Bounding box 5: $C_{xy0x0} \in [0.5, 1], C_{xy0x1} \in [0.5, 1], C_{xy1x0} \in [0.5, 1], C_{xy1x1} \in [0, 1]$

We can see that these constraints are easy to interpret and some intuitive results can be derived from them: For instance, from Bounding box 2, we can get that sustained oscillations exist if the values of C_{xy0x1} are close to each other and are sufficiently larger than any C_{xy0x0}, C_{xy1x0} and C_{xy1x1}.

5.6 Summary

In this chapter, we constructed a HGRN of the canonical repressilator. By computing and analyzing analytically a Poincaré map and based on a hypothesis, we develop a sufficient and necessary condition for the existence of sustained oscillations. Then we adapt the range enclosure property of Bernstein coefficients to find some bounding boxes in parameters space which satisfy this sufficient and necessary condition. These bounding boxes (intervals of parameters) can provide useful information for the design of synthetic circuits. Moreover, an intermediate result implies some new control strategies for sustained oscillations: Controlling the absolute values of the derivatives of one gene under certain regulations such that these values are sufficiently small, while keeping other parameters of the system unchanged. The work of this chapter can be considered as an extension of the limit cycle analysis method in Chapter 3 on a specific gene regulatory network because both works rely on the method of Poincaré map. The difference between the work of this chapter and the work of Chapter 3 is that, in the work of this chapter we analyze symbolically a HGRN while in the work of Chapter 3 the exact values of parameters are assumed to be known. This symbolic analysis approach can be further applied to other networks. This chapter focuses on the analysis of a HGRN of the canonical repressilator, while another important problem is the identification of parameters based on data which is investigated in the next chapter.

Chapter 6

Parameter Identification based on Time Series Data

In the previous chapter, we studied the dynamical properties of a specific HGRN under certain constraints on parameters. An another challenge is how to find these constraints on parameters from data. In this chapter, we study the problem of parameter identification of HGRNs. In the literature, there are works focusing on identifying parameters of HGRNs based on biological knowledge [72, 73]. This thesis studies parameter identification methods based on time series data. We consider the parameter identification problem as an optimization problem and we mainly focus on meta-heuristic optimization methods like the genetic algorithm. As opposed to many parameter identification problems, which aim to find a specific choice of parameters, in this chapter we want to find a set of parameters that can provide more information about the possible qualitative properties of the system. We use the HGRN model of the repressilator studied in the previous chapter as a case study, although the method can be generalized on other HGRNs.

6.1 Definition of the Parameter Identification Problem

This section defines the parameter identification problem concerned in this chapter. Firstly, the parameters and the form of the parameter set to identify are presented in Section 6.1.1. Then the type of data is presented in Section 6.1.2.

6.1.1 Representation of the Parameter Set to Identify

In this chapter, we consider the same HGRN studied in Chapter 5, but the method is generic and can be extended for other HGRNs. Chapter 5 focused on the analysis of qualitative behaviors so the absolute values of the thresholds that separate discrete levels

are not considered, while in this chapter, since the objective is to identify parameters from continuous expression data, we not only need to consider the parameters describing the time derivatives in each discrete state but also need to consider these thresholds.

Therefore, there are two classes of parameters considered in the work of this chapter: The parameters describing the time derivatives of each discrete state are denoted by V_{ac0a0}, V_{ac0a1}, V_{ac1a0}, V_{ac1a1}, V_{ba0b0}, V_{ba0b1}, V_{ba1b0}, V_{ba1b1}, V_{cb0c0}, V_{cb0c1}, V_{cb1c0}, V_{cb1c1}; and the parameters describing the thresholds are denoted by θ_{ca}, θ_{ab}, θ_{bc}. By assuming that all these parameters are positive, the dynamics of gene X is described by the following equations if X does not reach its maximum or minimum value.

$$\frac{dX}{dt} = \begin{cases} V_{XY0X0} & \text{if } x_Y < \theta_{YX} \wedge x_X < \theta_{XZ} \\ V_{XY0X1} & \text{if } x_Y < \theta_{YX} \wedge x_X \geq \theta_{XZ} \\ -V_{XY1X0} & \text{if } x_Y \geq \theta_{YX} \wedge x_X < \theta_{XZ} \\ -V_{XY1X1} & \text{if } x_Y \geq \theta_{YX} \wedge x_X \geq \theta_{XZ} \end{cases} \tag{6.1}$$

where $(Y, X, Z) \in \{(c, a, b), (a, b, c), (b, c, a)\}$; $x_a/x_b/x_c$ is the continuous expression of gene $a/b/c$ over time. If X reaches its maximum or minimum value, then its time derivative is 0 which corresponds to the existence of a sliding mode. In this chapter, we impose that the maximum value of each gene is 1 and the minimum value of each gene is 0 (the reason for this choice will be presented in the next subsection), so θ_{ca}, θ_{ab}, θ_{bc} are real numbers between 0 and 1. In fact, the HGRN studied in Chapter 5 can be considered as a normalization over two levels (between 0 and 2) of the model considered in this chapter.

Many works about parameter identification aim to find a specific choice of parameters, in other words, assign a specific value to each parameter. In the work of this chapter, instead of finding a specific choice of parameters, we want to find a parameter set which can be described by constraints on parameters under the following form:

$$\forall (Y, X) \in \{(c, a), (a, b), (b, c)\}, \forall i, j \in \{0, 1\}, V_{XYiXj} \in I_{XYiXj} = [\underline{I_{XYiXj}}, \overline{I_{XYiXj}}]$$

and

$$\forall (Y, X) \in \{(c, a), (a, b), (b, c)\}, \theta_{YX} \in I_{YX} = [\underline{I_{YX}}, \overline{I_{YX}}]$$

where $\forall (Y, X) \in \{(c, a), (a, b), (b, c)\}, \forall i, j \in \{0, 1\}, \underline{I_{XYiXj}}, \overline{I_{XYiXj}}, \underline{I_{YX}}, \overline{I_{YX}} \in \mathbb{R}, 0 \leq \underline{I_{XYiXj}} < \overline{I_{XYiXj}}, 0 \leq \underline{I_{YX}} < \overline{I_{YX}} \leq 1$. In other words, each parameter is constrained individually by an interval. The form of these constraints is the same as the form of separable constraints in Chapter 5. The difference is that we also consider here the constraints on

thresholds. In this chapter, this kind of parameter set constrained by intervals is called *regular parameter set*. A regular parameter set is denoted by P_s.

In order to simplify the parameter identification problem, we choose to fix the length of these intervals to identify; more precisely, we want to find a regular parameter set under the following form:

$\forall (Y,X) \in \{(c,a),(a,b),(b,c)\}, \forall i,j \in \{0,1\}, V_{XYiXj} \in$
$[\max(0, \hat{V}_{XYiXj} - \epsilon_{XYiXj}), \hat{V}_{XYiXj} + \epsilon_{XYiXj}]$

and

$\forall (Y,X) \in \{(c,a),(a,b),(b,c)\}, \theta_{YX} \in [\max(0, \hat{\theta}_{YX} - \epsilon_{YX}), \min(1, \hat{\theta}_{YX} + \epsilon_{YX})]$

where $\forall (Y,X) \in \{(c,a),(a,b),(b,c)\}, \forall i,j \in \{0,1\}$, ϵ_{XYiXj} and ϵ_{YX}, which are positive real numbers, are chosen (fixed), so we only need to identify the values of $\hat{V}_{XYiXj}$ and $\hat{\theta}_{YX}$.

The reason why we choose to identify a parameter set is that a parameter set can cover more possible qualitative properties of the system compared to a single choice of parameters, and the reason why we choose to use a regular parameter set is that it is easier to sample models from such a parameter set.

6.1.2 Data Representation

In this chapter, we investigate parameter identification methods based on time series data. The inference of gene regulatory networks from time series data is in fact an active research area [74, 75]. The time series data used in this chapter is formally represented as a sequence: $(x_X^{t_0}, x_X^{t_1}, ..., x_X^{t_m})$ where $X \in \{a,b,c\}$ represents one gene of the system; the sequence $(t_0, t_1, ..., t_m)$ is a sequence of time representing the moments when we know the continuous expressions of genes; $x_X^{t_i}$ represents the expression of gene X at time t_i.

For now, we do not have biological time series data that correspond to the canonical repressilator, thus simulated data is used in this chapter to verify the parameter identification methods. More precisely, we randomly choose parameters of this model and make a simulation from a randomly chosen initial state to obtain time series data. In this chapter, we simply choose that, for each gene, the maximum value is 1 and the minimum value is 0. Indeed, a priori, the maximum and minimum values are unknown; in that case, we can also consider these values as parameters to identify. In order to make the data more realistic, we also add noise independently on each data point following a normal distribution.

The data are presented in Table 6.1 and their illustration is given in Fig 6.1. In Table 6.1, there are some values which are less than 0 or bigger than 1 because of the added noise.

Table 6.1 – Time series data used to verify the parameter identification methods.

t_i	0	0.24	0.49	0.73	0.98	1.22	1.47	1.72	1.96	2.21	2.45	2.71	2.95	3.19	3.44	3.68	3.93	4.18	4.42	4.67	4.91
$x_a^{t_i}$	0.51	0.96	0.95	0.84	0.71	0.63	0.54	0.40	0.23	0.35	0.74	1.01	0.97	0.95	0.83	0.78	0.61	0.56	0.44	0.21	0.33
$x_b^{t_i}$	0.66	0.00	-0.02	0.00	-0.02	0.00	-0.02	-0.01	0.39	0.95	0.35	0.03	-0.02	0.00	-0.01	0.03	0.01	0.00	-0.01	0.36	1.03
$x_c^{t_i}$	0.82	0.62	0.92	1.00	0.99	0.98	0.98	1.01	0.90	0.44	0.00	0.26	0.67	0.94	0.99	1.01	1.01	0.97	1.00	0.88	0.43

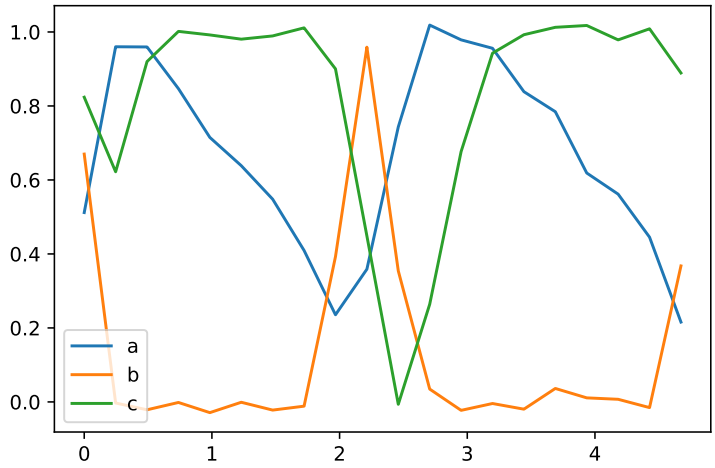


Figure 6.1 – Illustration of the time series data used to verify the parameter identification methods.

The parameters of the HGRN used to obtain these data are given as follows. Although these values are chosen randomly without any biological consideration, they can be used to verify the parameter identification methods at the first stage.

$$V_{ac0a0} = 1 \qquad V_{ac0a1} = 1.9 \qquad V_{ac1a0} = 1.3 \qquad V_{ac1a1} = 0.4$$
$$V_{ba0b0} = 3.8 \qquad V_{ba0b1} = 2.5 \qquad V_{ba1b0} = 2.7 \qquad V_{ba1b1} = 3.3$$
$$V_{cb0c0} = 1.5 \qquad V_{cb0c1} = 0.8 \qquad V_{cb1c0} = 1.9 \qquad V_{cb1c1} = 1.5$$
$$\theta_{ca} = 0.8 \qquad \theta_{ab} = 0.4 \qquad \theta_{bc} = 0.2$$

6.2 Description of the Method

6.2.1 Objective Function for the Optimization of Regular Parameter Set

We consider the parameter identification problem as an optimization problem. More precisely, we use an objective function to describe how well a regular parameter set fits the data and we want to find a regular parameter set that minimizes this objective function. This objective function is given in Eq (6.2).

$$\lim_{n_s \to \infty} \frac{1}{n_s} \sum_{M_i \in Sample(P_s)} f(M_i) \tag{6.2}$$

where n_s models are randomly sampled from a given regular parameter set P_s (here n_s models mean n_s different choices of parameters) following the uniform distribution. These n_s sampled models are denoted by $Sample(P_s) = \{M_1, M_2, ..., M_{n_s}\}$. The expression of function f is given in Eq (6.3).

$$f(M_i) = \frac{1}{3(m+1)} \sum_{X \in \{a,b,c\}} \sum_{j \in \{0,1,2,...,m\}} w(x_X^{t_j})(x_X^{i,t_j} - x_X^{t_j})^2 \tag{6.3}$$

where, for each sampled model M_i, we compute the simulated time series data from the same initial states ($x_a^{t_0}$, $x_b^{t_0}$, $x_c^{t_0}$), and these simulated data are denoted by $(x_X^{i,t_0}, x_X^{i,t_1}, ..., x_X^{i,t_m}), X \in \{a, b, c\}$. The mean squared error is used to measure the difference between the input data and these simulated data. We also add weights to the mean squared error because some data points are more important than others for parameter identification. The weight of data point $x_X^{t_j}$ is denoted by $w(x_X^{t_j})$, which is proportional to the estimated time derivative of $x_X^{t_j}$. More precisely, $w(x_X^{t_j}) = \max(|\frac{x_X^{t_j} - x_X^{t_{j-1}}}{t_j - t_{j-1}}|, |\frac{x_X^{t_{j+1}} - x_X^{t_j}}{t_{j+1} - t_j}|)$ if $0 < j < m$, $w(x_X^{t_j}) = |\frac{x_X^{t_{j+1}} - x_X^{t_j}}{t_{j+1} - t_j}|$ if $j = 0$ and $w(x_X^{t_j}) = |\frac{x_X^{t_j} - x_X^{t_{j-1}}}{t_j - t_{j-1}}|$ if $j = m$. In fact, this objective function describes the average of function f on a regular parameter set.

In practice, it is impossible to let n_s converge to infinity, so in the implementation we choose a finite value of n_s to compute the estimated value of the objective function.

6.2.2 Estimation of Potential Parameter Set

The final objective of this work is to find a regular parameter set that optimizes globally this objective function (a global minimum). However, finding the global minimum is non-

trivial. So what we can do is to find some local minimums.

In the parameter space, there exist some local minimums which are not reasonable, see for example Fig 6.2. This local minimum is not reasonable because the frequencies of the models that belong to this local minimum are too high. Such unreasonable local minimum exists because there is a finite number of data points in the real sequence. The number of such unreasonable local minimums decreases when the number of data points increases.

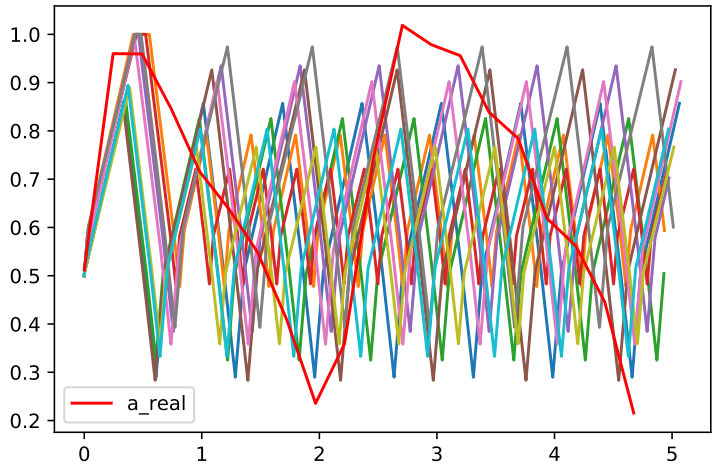


Figure 6.2 – Illustration of the real sequence of gene A and some simulated sequences of gene A by models belonging to a local minimum which is not reasonable.

In order to avoid finding these unreasonable local minimums, before applying the optimization algorithms, we propose a method to estimate a potential parameter set that does not contain these unreasonable local minimums. This idea is shown in Fig 6.3. The grey area represents the whole parameter space, the red squares represent some regular parameter sets which are local minimums. In this illustration, for example, the left red square with the illustration of simulated data represents an unreasonable local minimum and the right one with the illustration of simulated data represents a reasonable local minimum. What we want to do is to find a potential parameter set (the blue area) that does not contain the unreasonable local minimums. Then we could search for local minimums starting from this potential parameter set, which could improve the efficiency of the searching process.

Two types of constraints are considered to compute this potential parameter set. The

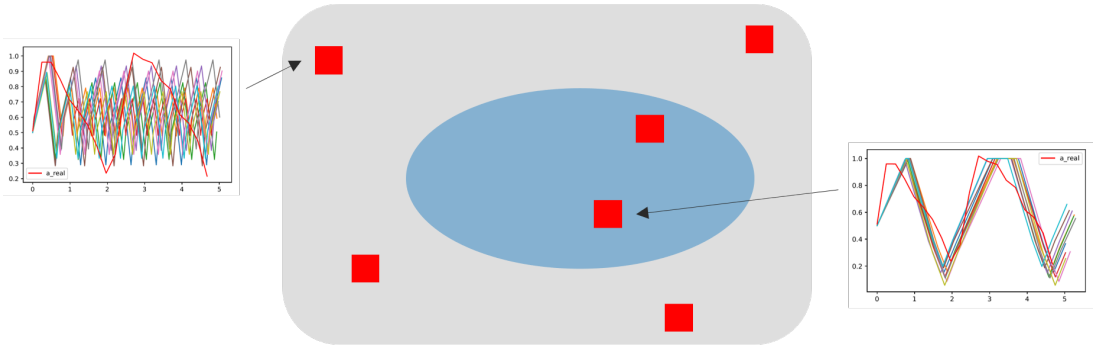


Figure 6.3 – Illustration of the idea to estimate a potential parameter set that does not contain the unreasonable local minimums. The grey area represents the whole parameter space, the red squares represent some regular parameter set which are local minimums and the blue area represents this potential parameter set.

first type consists in constraints on the maximum absolute value of the time derivatives of genes, which can be computed by the estimated time derivatives. More precisely, the maximum absolute value of the estimated time derivative of one gene is increased by an order of magnitude and this increased value is used as the constraint on the maximum absolute value of the time derivatives of this gene. The second type is based on the number of boundaries reached between t_0 and t_m. We assume that this sequence corresponds to an oscillation, and that it shows more than one period and less than two periods. So, based on the analysis of this HGRN in Chapter 5, we know that the number of boundaries reached between t_0 and t_m must be between 6 and 36, where 6 is the minimum number of boundaries reached by one period and the number of boundaries reached by two periods must be less than 36. Details about how this potential parameter set is used are presented in the following section.

6.2.3 Parameter Identification based on Genetic Algorithm

Generally, the methods to optimize an objective function can be classified in two categories based on whether the information of gradient is used or not. For now, we have not yet found a method to compute the gradient of this objective function, so we chose to use methods that are not based on gradient.

The most used class of optimization methods that is not based on gradient is meta-heuristics [76–82]. Among different meta-heuristics, in this chapter, we choose to use the genetic algorithm [83–85] which is one of the most studied meta-heuristics. In fact,

genetic algorithms have also been used to identify parameters of HGRNs in [73], which, contrary to this work, uses biological knowledge to identify the parameters. The idea of genetic algorithm comes from the process of natural selection. The main pipeline of genetic algorithm is illustrated in Fig 6.4. Details about each step are presented as follows.

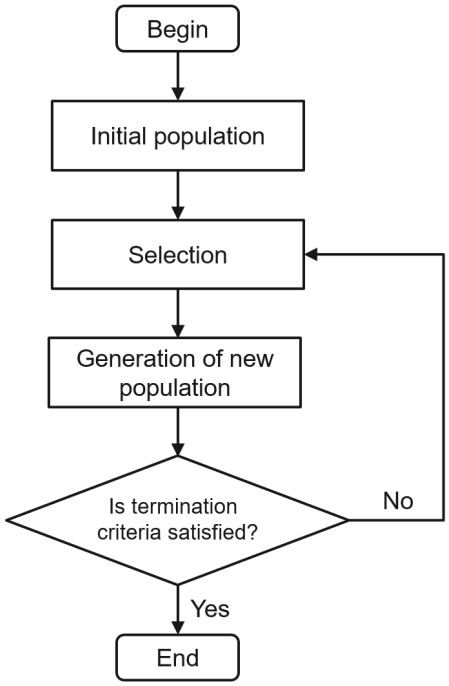


Figure 6.4 – Illustration of the major steps of a genetic algorithm.

In the step of initial population, some regular parameter sets that intersect the potential parameter set are randomly generated. In practice, to verify if a regular parameter set intersects the potential parameter set, we simply verify if the center of this regular parameter set (for example, if $V_{XYiXj} \in [\max(0, \hat{V}_{XYiXj} - \epsilon_{XYiXj}), \hat{V}_{XYiXj} + \epsilon_{XYiXj}]$, then the center of V_{XYiXj} is $\hat{V}_{XYiXj}$) belongs to the potential parameter set. These randomly generated regular parameter sets are called the initial population (the first generation).

In the step of selection, we compute the estimated values (as stated before, the exact values of the objective function are hard or maybe impossible to compute, so we estimate the values by choosing a finite value of n_s) of the objective function of each regular parameter set in the current population and choose the top n_{top} regular parameter sets, which are the n_{top} regular parameter sets with the minimum n_{top} estimated values of the objective function.

In the step of generation of new population (new generation), we compute the new

population based on the top n_{top} regular parameter sets from the previous population. The generation of new population is based on two operators: Crossover and mutation. In the literature, many sorts of crossover and mutation have been proposed [86]. In this chapter, we adapt the classic crossover and mutation which are presented as follows.

Consider two regular parameter sets P_s^p and P_s^q. The intervals of P_s^p and P_s^q are denoted by

$$V_{XYiXj} \in [\max(0, \hat{V}^p_{XYiXj} - \epsilon_{XYiXj}), \hat{V}^p_{XYiXj} + \epsilon_{XYiXj}]$$

$$\theta_{YX} \in [\max(0, \hat{\theta}^p_{YX} - \epsilon_{YX}), \min(1, \hat{\theta}^p_{YX} + \epsilon_{YX})]$$

and

$$V_{XYiXj} \in [\max(0, \hat{V}^q_{XYiXj} - \epsilon_{XYiXj}), \hat{V}^q_{XYiXj} + \epsilon_{XYiXj}]$$

$$\theta_{YX} \in [\max(0, \hat{\theta}^q_{YX} - \epsilon_{YX}), \min(1, \hat{\theta}^q_{YX} + \epsilon_{YX})]$$

respectively, where $(Y, X) \in \{(c, a), (a, b), (b, c)\}, i, j \in \{0, 1\}$. The operator crossover can generate a new regular parameter set, denoted by P_s^{pq}, from P_s^p and P_s^q. The intervals of P_s^{pq} are

$$V_{XYiXj} \in [\max(0, \hat{V}^{pq}_{XYiXj} - \epsilon_{XYiXj}), \hat{V}^{pq}_{XYiXj} + \epsilon_{XYiXj}]$$

$$\theta_{YX} \in [\max(0, \hat{\theta}^{pq}_{YX} - \epsilon_{YX}), \min(1, \hat{\theta}^{pq}_{YX} + \epsilon_{YX})]$$

where

$$\hat{V}^{pq}_{XYiXj} = \alpha \hat{V}^p_{XYiXj} + (1 - \alpha)\hat{V}^q_{XYiXj}$$

$$\hat{\theta}^{pq}_{YX} = \alpha \hat{\theta}^p_{YX} + (1 - \alpha)\hat{\theta}^q_{YX}$$

where α is a randomly generated real number from $[0, 1]$ following a uniform distribution.

The mutation operator can generate a new regular parameter set $P_s^{p\prime}$ from P_s^p. The intervals of $P_s^{p\prime}$ are

$$V_{XYiXj} \in [\max(0, \hat{V}^{p\prime}_{XYiXj} - \epsilon_{XYiXj}), \hat{V}^{p\prime}_{XYiXj} + \epsilon_{XYiXj}]$$

$$\theta_{YX} \in [\max(0, \hat{\theta}^{p\prime}_{YX} - \epsilon_{YX}), \min(1, \hat{\theta}^{p\prime}_{YX} + \epsilon_{YX})]$$

where

$$\hat{V}^{p\prime}_{XYiXj} = \max(0, \beta_{XYiXj}\hat{V}^{p}_{XYiXj})$$

$$\hat{\theta}^{p\prime}_{YX} = \min(1, \max(0, \beta_{YX}\hat{\theta}^{p}_{YX}))$$

where $\beta_{XYiXj}, \beta_{YX}$ are randomly generated real numbers from $[1-\delta, 1+\delta]$ where δ is a small positive real number. In other words, the mutation operator adds some small noises on the original intervals.

In order to compute the new population, firstly we apply the crossover on pairs of top n_{top} regular parameter sets from the previous populations to get some new regular parameter sets, and then we apply the mutation on these new regular parameter sets to get finally the new population.

Different criteria can be used to determine when the algorithm stops; in this work, we choose to fix the number of generations, in other words the loop in Fig 6.4 is repeated for a fixed number of times.

6.3 Application

The minimum value of the objective function of each generation (the estimated value of the objective function of the regular parameter set that minimizes the objective function in the current generation) is shown in Fig 6.5. The execution time of the algorithm is 17 minutes. Computations were performed on a standard laptop computer, with an Intel Core I7-8550U 1.80GHz processor and 16.0GB RAM. Globally, we can see that new generations are generally better with regard to the estimated values of the objective function and it seems that this curve converges. So we can assume that a local minimum is found which means that the best regular parameter set in the last population is very close to a local minimum.

Simulated data of this best regular parameter set in the last population are shown in Fig 6.6. We can see that the simulated data are close to the input data, which indicates that this parameter set could potentially cover the real qualitative properties of the system.

This regular parameter set is given as follows. We can see that this regular parameter set does not include the real parameters. In fact, because of the facts that we search for a set of parameters instead of a specific choice of parameters and noises are added to data,

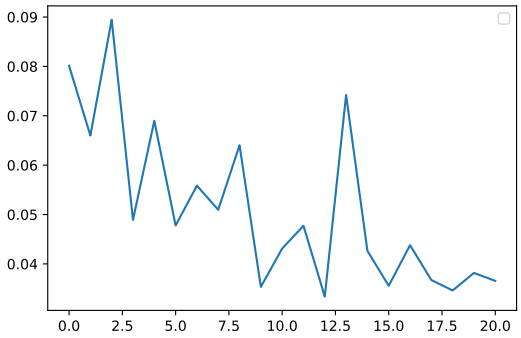


Figure 6.5 – The training curve of the genetic algorithm. Abscissa represents the number of generations and ordinate represents the minimum value of the objective function of each generation.

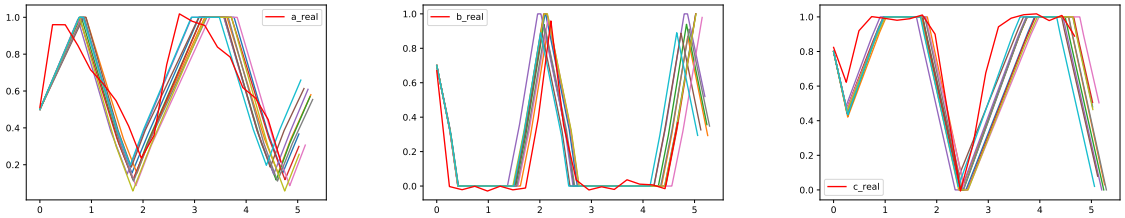


Figure 6.6 – Illustration of the last regular parameter set found by the genetic algorithm. Red curves represent input data and other curves represent simulated data of this regular parameter set.

even the global minimum does not necessarily include the real parameters.

$$
\begin{array}{llll}
V_{ac0a0} \in [0.66, 0.76] & V_{ac0a1} \in [0.58, 0.68] & V_{ac1a0} \in [0.67, 0.77] & V_{ac1a1} \in [0.89, 0.99] \\
V_{ba0b0} \in [1.45, 1.55] & V_{ba0b1} \in [1.74, 1.84] & V_{ba1b0} \in [2.00, 2.10] & V_{ba1b1} \in [1.41, 1.51] \\
V_{cb0c0} \in [0.73, 0.83] & V_{cb0c1} \in [0.63, 0.73] & V_{cb1c0} \in [1.29, 1.39] & V_{cb1c1} \in [1.36, 1.46] \\
\theta_{ca} \in [0.87, 0.97] & \theta_{bc} \in [0.27, 0.37] & \theta_{ab} \in [0.29, 0.39] &
\end{array}
$$

The above result is only the result of one execution of the algorithm. By executing the algorithm for several times, different regular parameter sets (local minimums) can be found. Some regular parameter sets can generate data that are visually close to the input data, while others cannot. This leads to an open question of this work: For a local minimum found by this method which cannot generate data that visually fit the input

data, does there exist another local minimum which generates data that visually fit better the input data? Or can we verify if the found local minimum is the global minimum? We note that the global minimum does not necessarily generate data that visually fit best the input data, but we can assume that it is true if the objective function is chosen properly.

6.4 Summary

In this chapter, we propose a method based on a genetic algorithm to identify parameter sets from time series data. The results show that the obtained parameter sets could potentially cover the qualitative properties of the real system. The major limit of this work is that only simulated data are used. To further evaluate this method, real data from biological experiments are required, because the distribution of real data could be different from simulated data which could potentially influence the effectiveness of the method. There are also open questions related to the identification of the global minimum that can be further investigated. Generally, this work is only a beginning of the study of parameter identification of HGRN based on time series.

The study of this chapter and the previous one were based on the 3-dimensional repressilator. In the next chapter, we will investigate the dynamical properties of the 4-dimensional repressilator with a discrete formalism.

Chapter 7

Condition for a Discrete Periodic Attractor in 4-dimensional Repressilators

In this chapter, we study the 4-dimensional repressilators, which is a continuation of the work in Chapter 5. Initially, the idea was to develop, like in Chapter 5, constraints on celerities of 4-dimensional repressilators for the existence of sustained oscillations. However, it is a non-trivial task, because this extra dimension makes the use of Poincaré map to analyze the stability of characteristic states more complicated. Thus, in this chapter, we turn to discrete models and aim to find the condition for the existence of a discrete periodic attractor based on the asynchronous semantics, because the existence of a discrete periodic attractor based on the asychronous semantics in a discrete model is a sufficient condition for the existence of a limit cycle in the associated HGRN. We aim to find such a condition described by the topological features of influence graphs, which follows some theoretical works [87–91]. Our major contributions include: 1) discovering that, with one exception, the relations between gene regulation thresholds do not impact the existence of discrete periodic attractors in any of the influence graphs considered in this study; 2) identifying a sufficient and necessary condition of simple form for the existence of a discrete periodic attractor when the same exception is ignored; 3) identifying new topological features of influence graphs that are necessary for predicting the existence of discrete periodic attractor in 4-dimensional repressilators. The work of this chapter has been presented at CMSB 2023 as regular paper [25].

7.1 4-dimensional Repressilators

In the literature, there are works studying extensions of the 3-dimensional canonical repressilator by adding more genes into the network [92–96]. Most of these works focus on

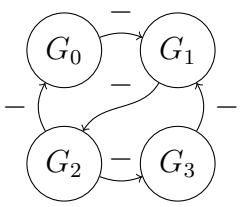


Figure 7.1 – An influence graph of a 4-dimensional repressilator.

some specific networks. In this work, we study all 4-dimensional gene regulatory networks where genes are linked only through inhibition, and where every gene has an impact on at least one other gene. These networks are called 4-dimensional repressilators. An influence graph of such networks is shown in Fig 7.1.

Our analysis of these 4-dimensional repressilators is based on two underlying assumptions about their dynamics.

Assumption 7 *If one gene influences m different genes, then it has m distinct thresholds that correspond to each of these m genes.*

Consider the influence graph of Fig 7.1, G_2 inhibits G_0 and G_3, so G_2 has two distinct thresholds because of Assumption 7, meaning that it has three discrete levels: $0, 1, 2$. Since each of the other genes only influences one other gene, they only have two discrete levels: $0, 1$. A similar assumption can be found in [34] for example.

The relations between the two thresholds of G_2 can have impact on the dynamical properties of the system. To show the relations between thresholds on an influence graph, we introduce the notion of influence graph with thresholds which is defined as $IGS = (V, A, s)$ where V and A are the sets of genes (nodes) and regulations (edges) between genes respectively, as in the definition of an influence graph (see Chapter 2), and the function s assigns an integer to each regulation that represents the minimum discrete level of the source gene necessary to inhibit the target gene. Thus, the function s also characterizes the relationship between thresholds.

From the influence graph of Fig 7.1, by considering all different relations between thresholds, we can get two different influence graphs with thresholds as illustrated in Fig 7.2. For the regulation $G_2 \to G_3$ in the left influence graph with thresholds, $s(G_2 \to G_3) = 2$ (which is the number on the arc) means that G_3 is inhibited by G_2 if the discrete level of G_2 is bigger or equal to 2. For the regulation $G_2 \to G_0$ in the same graph, $s(G_2 \to G_0) = 1$ means that G_0 is inhibited by G_2 if the discrete level of G_2 is bigger or

Figure 7.2 – Different influence graphs with thresholds, corresponding to the influence graph of Fig 7.1.

equal to 1. We can see that, in the left influence graph with thresholds, the threshold of G_2 triggering the inhibition of G_0 is smaller than the threshold triggering the inhibition of G_3, while in the right influence graph with thresholds, the situation is reversed. We can also see that the function s gives all possible discrete levels of the system.

A priori, different discrete models (logic programs) can be associated to the same influence graph with thresholds, particularly when one gene is inhibited by several genes. Moreover, different choices of logic programs lead to different transition graphs, in other words different dynamical properties. In this chapter, we make an assumption about the dynamics when one gene is inhibited by several genes.

Assumption 8 *In an influence graph with threshold $IGS = (V, A, s)$, for any gene G, its discrete level can decrease by 1 if there exists a regulation from G' to G and the current discrete level of G' is bigger or equal to $s(G' \to G)$, otherwise its discrete level can increase by 1.*

In fact, Assumption 8 is equivalent to assume that the inhibitions are disjunctive, meaning that only one inhibitor is enough to decrease the target gene. Similar assumptions about the disjunction or conjunction of gene regulation can be found in [97, 98]. Consider the influence graph with thresholds on the left of Fig 7.2; Assumption 8 leads to the following

logic program (for reminder, the discrete level of gene G_i is noted by a_i):

$$\begin{aligned}
&a_0 = 0 \leftarrow (a_2 \geq 1)\\
&a_0 = 1 \leftarrow (a_2 < 1)\\
&a_1 = 0 \leftarrow (a_0 \geq 1) \vee (a_3 \geq 1)\\
&a_1 = 1 \leftarrow (a_0 < 1) \wedge (a_3 < 1)\\
&a_2 = 0 \leftarrow (a_1 \geq 1) \wedge (a_2 \leq 1)\\
&a_2 = 1 \leftarrow (a_1 \geq 1) \wedge (a_2 = 2)\\
&a_2 = 1 \leftarrow (a_1 < 1) \wedge (a_2 = 0)\\
&a_2 = 2 \leftarrow (a_1 < 1) \wedge (a_2 \geq 1)\\
&a_3 = 0 \leftarrow (a_2 \geq 2)\\
&a_3 = 1 \leftarrow (a_2 < 2)
\end{aligned} \tag{7.1}$$

Note that, for instance, there needs to be two rules in order to make G_2 increase to the expression level 2: One to update it from level 0 to level 1 (line 7) and one to make it increase from 1 to 2 (line 8); this is because we did not constraint the dynamics to be unitary and we thus need to encode this property inside the rules. Using Assumption 8, we get a unique discrete model from any influence graph with thresholds, which simplifies the analysis.

In this work, we use the asynchronous semantics, which is formally defined as follows. We consider a system with N genes noted as $G_0, G_1, G_2, ..., G_N$. For any discrete state d_s at time t, if there exists a logic rule $a_{i_0} = k \leftarrow \phi$ where $i_0 \in \{0, 1, 2, ..., N\}$, such that d_s satisfies ϕ and $d_s^{i_0} \neq k$, then at time $t + 1$, the system can reach the new discrete state d'_s such that $d'^i_s = d^i_s$ for $i \in \{0, 1, 2, ..., N\} \setminus \{i_0\}$ and $d'^{i_0}_s = k$. The transition graph of discrete states based on the asynchronous semantics corresponding to the discrete model of Eq (7.1) is illustrated in Fig 7.3.

The general logic rules for an arbitrary IGS are given in Eq (7.2).

$$\begin{aligned}
&a_i = k + 1 \leftarrow (a_i = k) \wedge (k < Max_{G_i}) \wedge (\forall G \in reg(G_i), G < s(G \rightarrow G_i))\\
&a_i = k - 1 \leftarrow (a_i = k) \wedge (k > 0) \wedge (\exists G \in reg(G_i), G \geq s(G \rightarrow G_i))
\end{aligned} \tag{7.2}$$

where Max_{G_i} is the maximum discrete level of G_i and $reg(G_i)$ is the set of all genes that inhibit G_i. Obviously, the rules of Eq (7.1) can be derived from the ones of Eq (7.2) by simplification. Some simplifications also involve the knowledge of the asynchronous

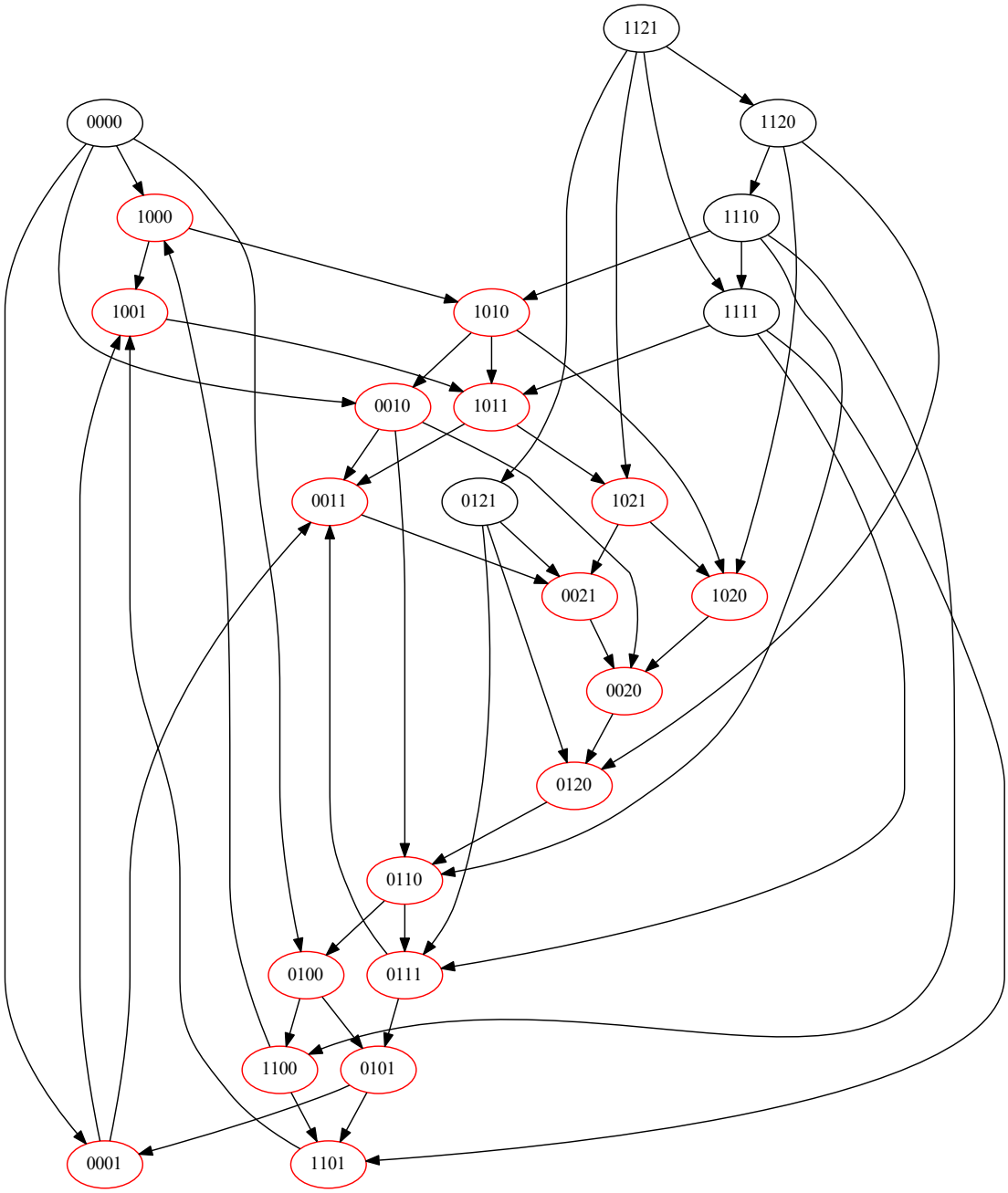


Figure 7.3 – Transition graph of discrete states of a model of 4-dimensional repressilator (corresponding to Eq (7.1)) based on asynchronous semantics. Red discrete states represent a discrete periodic attractor.

semantics given in the previous paragraph.

7.2 Feature Selection and Search of Candidate Condition based on Decision Tree

The definition of a discrete periodic attractor is given as follows.

Definition 15 *A discrete periodic attractor is a set of discrete states E_a such that:*

- *E_a contains at least two discrete states.*
- *For any two discrete states $d_s, d'_s \in E_a$, there exists a path in the transition graph of discrete states from d_s to d'_s.*
- *For any discrete state $d_s \in E_a$ and for any discrete state $d'_s \notin E_a$, there is no path in the transition graph of discrete states from d_s to d'_s.*

One example of discrete periodic attractor is given by the red discrete states in Fig 7.3.

In order to find a condition for the existence of a discrete periodic attractor, we firstly construct a decision tree model following the five steps below. The reason why we want to construct a decision tree model is that if we could obtain a decision tree with a classification accuracy of 1.0 to predict the existence of a discrete periodic attractor, then this decision tree is equivalent to a sufficient and necessary condition for the existence of a discrete periodic attractor in 4-dimensional repressilators. This work indeed only considers a finite number of discrete models, and a decision tree can be intuitively explained.

1. Generate all influence graphs with thresholds of 4-dimensional repressilators.
2. For each influence graph with thresholds, check the existence of a discrete periodic attractor using an attractor identification algorithm. Here, an influence graph having a discrete periodic attractor means that the associated discrete model (which is unique because of Assumption 8) has a discrete periodic attractor.
3. Compute manually some features which could be potentially used to predict the existence of a discrete periodic attractor.
4. Construct a decision tree model which uses the features in the previous step to predict the existence of a discrete periodic attractor.
5. Manually drop some features which do not influence the prediction result.

In step 1, there are 50625 influence graphs with thresholds in total without removing the graphs that are equivalent.

In step 2, we use the function *attracting_components* of the Python library NetworkX to verify the existence of a discrete periodic attractor. We find that any influence graph with thresholds which has a discrete periodic attractor has only one discrete periodic attractor.

In step 3, we compute two classes of features on the influence graph to describe the topology of the influence graph: The *number of cycles of length n* and the *total out-degree of cycles of length n*.

For the first class, since the system has 4 genes, there are only cycles of length 2, 3 and 4. We use $C2$, $C3$ and $C4$ to represent the numbers of cycles of length 2, 3 and 4, respectively. For example, for the influence graph in Fig 7.1, there is no cycle of length 2 or 4 ($C2 = 0, C4 = 0$) and two cycles of length 3 ($C3 = 2$). It is logical to use these features to predict the existence of a discrete periodic attractor because, in this class of repressilators, the length of a cycle determines whether it is a negative feedback loop or a positive feedback loop as there are only inhibition regulations and the presence of loops is related to the existence of attractor(s). For example, it is already known that the presence of a negative feedback loop is a necessary condition for sustained oscillations [89] and the presence of positive feedback loop is a necessary condition for multistability [88].

The second class of features is a new class of features introduced in this work which is defined formally as follows.

Definition 16 (Total out-degree of cycles of length n) *The total out-degree of cycles of length n is the total number of arcs which go from a vertex which belongs to a cycle of length n to a vertex which does not belong to this cycle.*

For example, for the influence graph in Fig 7.1, the arc $G_2 \to G_3$ goes from the cycle of length 3: $G_0 \to G_1 \to G_2$, to G_3, which does not belong to this cycle. There are two arcs like this in this influence graph: $G_2 \to G_3$ (for the cycle $G_0 \to G_1 \to G_2$) and $G_2 \to G_0$ (for the cycle $G_1 \to G_2 \to G_3$). So the total out-degree of cycles of length 3 is 2. Since, in this influence graph, there is no cycle of length 2, the total out-degree of cycles of length 2 is 0. Since any graph considered in this chapter has only 4 genes, the total out-degree of cycles of length 4 is always 0. We use $OD2$ and $OD3$ to denote the total out-degree of cycles of length 2 and 3, respectively.

To explain the motivation about these features describing the total out-degree of cycles, let's consider the two influence graphs in Fig 7.4. The dynamical properties of these two influence graphs are different: Any influence graph with thresholds associated

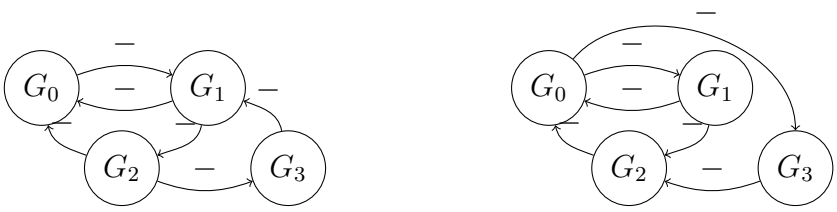


Figure 7.4 – Left: Influence graph always having a discrete periodic attractor. Right: Influence graph never having a discrete periodic attractor.

to the left influence graph has a discrete periodic attractor while any influence graph with threshold associated to the right influence graph does not have a discrete periodic attractor. However, the topologies of these two influence graphs are similar: The numbers of cycles of length 2, 3 and 4 of these two graphs are identical and they both have 6 arcs. In order to find a condition for discrete periodic attractor, we need to find a way to exhibit the topological difference between these two graphs, and these new features are effective: For the left graph, $OD2 = 1, OD3 = 3$; for the right one, $OD2 = 2, OD3 = 2$. Note that these features do not depend on the relations between thresholds.

In step 4, we construct a decision tree to predict the existence of a discrete periodic attractor based on the 5 features $C2$, $C3$, $C4$, $OD2$ and $OD3$ using all influence graphs with thresholds considered in this chapter. This decision tree is constructed automatically using the decision tree model of the Python library Scikit-learn.

The accuracy of prediction of this decision tree is nearly 0.9990. Initially, we wished this accuracy to be 1 because in that case, the decision tree would provide a sufficient and necessary condition for the existence of a discrete periodic attractor. This small lack of accuracy is actually caused by a few influence graphs with thresholds all related to the same influence graph. By analyzing this influence graph, a very interesting result arises:

There exists one particular influence graph such that, for any influence graph with thresholds that is not associated to this influence graph (or any isomorphism), the existence of a discrete periodic attractor does not depend on the relations between thresholds and can be predicted by this decision tree with an accuracy of 1.

This particular influence graph is shown in Fig 7.5. This figure also presents all influence graphs with thresholds, associated to this influence graph, having a discrete periodic attractor. In fact, with the exception of the relation between the thresholds presented in this figure (the only arcs assigned with numbers), the relations between the thresholds of G_2 do not influence the existence of a discrete periodic attractor, meaning that for any

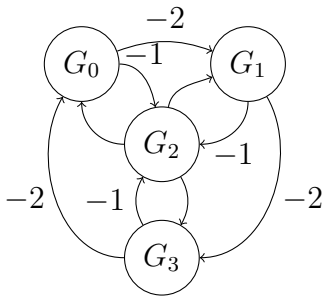


Figure 7.5 – The particular influence graph whose the relations between thresholds influence the existence of a discrete periodic attractor. Amongst all influence graphs with thresholds associated to this influence graph, only a subset has a discrete periodic attractor; this subset is characterized by the thresholds depicted in the figure.

order of thresholds of G_2 added to this figure, it always has a discrete periodic attractor.

In step 5, we manually drop features that do not influence the accuracy of the decision tree model. To do so, we re-train the decision tree each time after dropping one feature and observe if the accuracy decreases. Finally, we find that only keeping the features $OD2$ and $OD3$ ensures the same accuracy. We have also verified that this is the only couple of features that can maintain this accuracy. The final decision tree is shown in Fig 7.6. In this tree, the blue leaves predict the existence of a discrete periodic attractor, and the other leaves predict the non-existence of a discrete periodic attractor. We can see that except the second leaf from the right, models in all other leaves are classified correctly. In fact, this second leaf from the right contains all models associated to the particular influence graph of Fig 7.5. This means that apart from this particular influence graph, this tree describes a sufficient and necessary condition for the existence of a discrete periodic attractor in 4-dimensional repressilators.

7.3 Condition Simplification

In this subsection, we compute a simplified sufficient and necessary condition for the existence of a discrete periodic attractor based on the decision tree of Fig 7.6. The four paths which end at blue leaves, which are the leaves related to the existence of a discrete

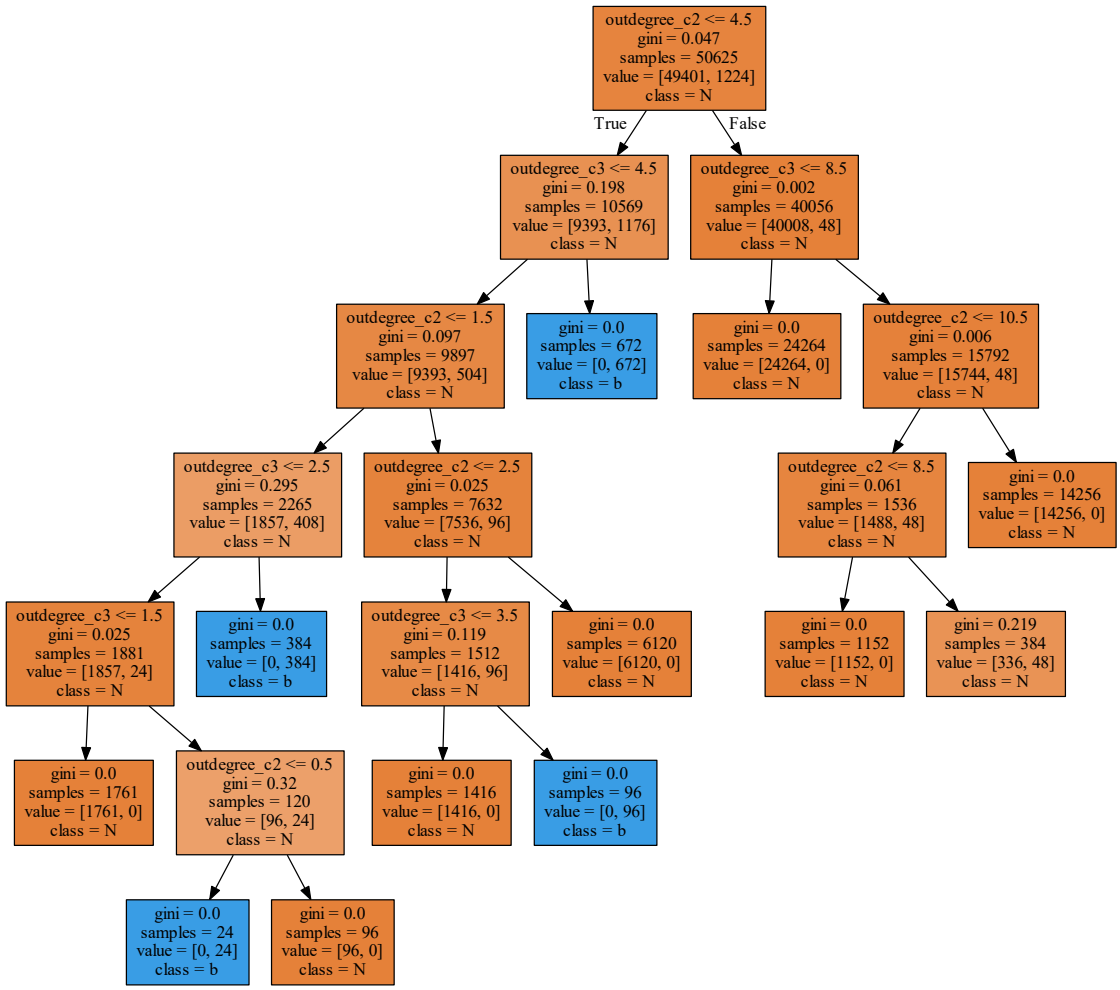


Figure 7.6 – A decision tree model to predict the existence of a discrete periodic attractor. Blue leaves represent the models classified as having a discrete periodic attractor and orange leaves represent the models classified as not having a discrete periodic attractor. "gini" describes the purity of models in a node regarding the two classes considered here: Models having a discrete periodic attractor and models not having a discrete periodic attractor; if all models in a node belong to the same class then gini $= 0$, otherwise gini > 0 ($gini = 1 - (\frac{number_{class1}}{number_{total}})^2 - (\frac{number_{class2}}{number_{total}})^2$). "sample" represents the number of models in a node. The first value of "value" represents the number of models not having a discrete periodic attractor and the second value of "value" represents the number of models having a discrete periodic attractor.

periodic attractor, are equivalent to the following logic rules:

$$\begin{aligned}(OD2 \leq 4 \wedge OD3 \leq 4 \wedge OD2 \leq 1 \wedge OD3 \leq 2 \wedge OD3 > 1 \wedge OD2 \leq 0) \vee \\ (OD2 \leq 4 \wedge OD3 \leq 4 \wedge OD2 \leq 1 \wedge OD3 > 2) \vee \\ (OD2 \leq 4 \wedge OD3 \leq 4 \wedge OD2 > 1 \wedge OD2 \leq 2 \wedge OD3 > 3) \vee \\ (OD2 \leq 4 \wedge OD3 > 4)\end{aligned} \tag{7.3}$$

Since $OD2$ and $OD3$ are integers, these logic rules can be simplified as follows.

$$\begin{aligned}(OD2 = 0 \wedge OD3 = 2) \vee \\ (OD2 \in \{0, 1\} \wedge OD3 \in \{3, 4\}) \vee \\ (OD2 = 2 \wedge OD3 = 4) \vee \\ (OD2 \in \{0, 1, 2, 3, 4\} \wedge OD3 \in \{5, 6, 7, ...\})\end{aligned} \tag{7.4}$$

Moreover, for all influence graphs, $OD2$ and $OD3$ are not independent and they are linked by the following constraints. Since there is a finite number of models, these constraints can be easily obtained by enumerating all models and comparing the values of $OD2$ and $OD3$.

$$\begin{aligned}\text{If } OD2 = 0 \text{ then } OD3 \in \{0, 1, 2, 3\} \\ \text{If } OD2 = 1 \text{ then } OD3 \in \{0, 1, 2, 3, 4\} \\ \text{If } OD2 = 2 \text{ then } OD3 \in \{0, 1, 2, 3, 4, 5\} \\ \text{If } OD2 = 3 \text{ then } OD3 \in \{0, 1, 2, 3\} \\ \text{If } OD2 = 4 \text{ then } OD3 \in \{0, 1, 2, 3, 4, 6, 8\}\end{aligned} \tag{7.5}$$

By combining Eq (7.4) and Eq (7.5), we get the following result, which is a sufficient and necessary condition for the existence of a discrete periodic attractor in case that the influence graph is not equivalent to the one in Fig 7.5:

$$\begin{aligned}(OD2 = 0 \wedge OD3 \in \{2, 3\}) \vee \\ (OD2 = 1 \wedge OD3 \in \{3, 4\}) \vee \\ (OD2 = 2 \wedge OD3 \in \{4, 5\}) \vee \\ (OD2 = 4 \wedge OD3 \in \{6, 8\})\end{aligned} \tag{7.6}$$

This result is of simple form and we can also find some patterns in it: The values of $OD2$ are powers of 2 ($2^0, 2^1, 2^2$) except 0, and $OD3$ increases as $OD2$ increases. These patterns might lead to some general theoretical results for N-dimensional repressilators.

7.4 Number of Oscillatory Dimensions in a Discrete Periodic Attractor

In this subsection, we also investigate the number of oscillatory dimensions in the discrete periodic attractors. For a discrete periodic attractor, oscillating in 3 dimensions means that there exists one dimension i_0 and an integer a such that for any discrete state d_s in this discrete periodic attractor, $d_s^{i_0} = a$, and for any dimension i which differs from i_0, we can find two discrete states d_{s_1}, d_{s_2} in this discrete periodic attractor, such that $d_{s_1}^i \neq d_{s_2}^i$. An example of a discrete periodic attractor that oscillates in 3 dimensions is given in Fig 7.7 where there is no oscillation in the first dimension. Oscillating in 4 dimensions means that for any dimension i, we can find two discrete states d_{s_1}, d_{s_2} in this discrete periodic attractor, such that $d_{s_1}^i \neq d_{s_2}^i$. For example, the discrete periodic attractor in Fig 7.3 oscillates in 4 dimensions.

By automatically verifying the isomorphisms of all influence graphs with discrete periodic attractors except the influence graph of Fig 7.5, we find that there are, in total, only 8 different (non-isomorphic) influence graphs which always have discrete periodic attractors. Any influence graph with thresholds corresponding to these 8 influence graphs has only one discrete periodic attractor. Among these 8 influence graphs, 2 of them (Fig 7.8) can have discrete periodic attractors which oscillate in both 3 and 4 dimensions depending on different relations between thresholds, and the other 6 (Fig 7.9) only have discrete periodic attractors which oscillate in 4 dimensions.

7.5 Summary

In this chapter, we study the condition for the existence of a discrete periodic attractor in 4-dimensional repressilators under some dynamical assumptions. With the guide of decision tree models, we find a special influence graph for which the relations between thresholds influence the existence of a discrete periodic attractor. For all other influence graphs, we show that the existence of a discrete periodic attractor does not depend on

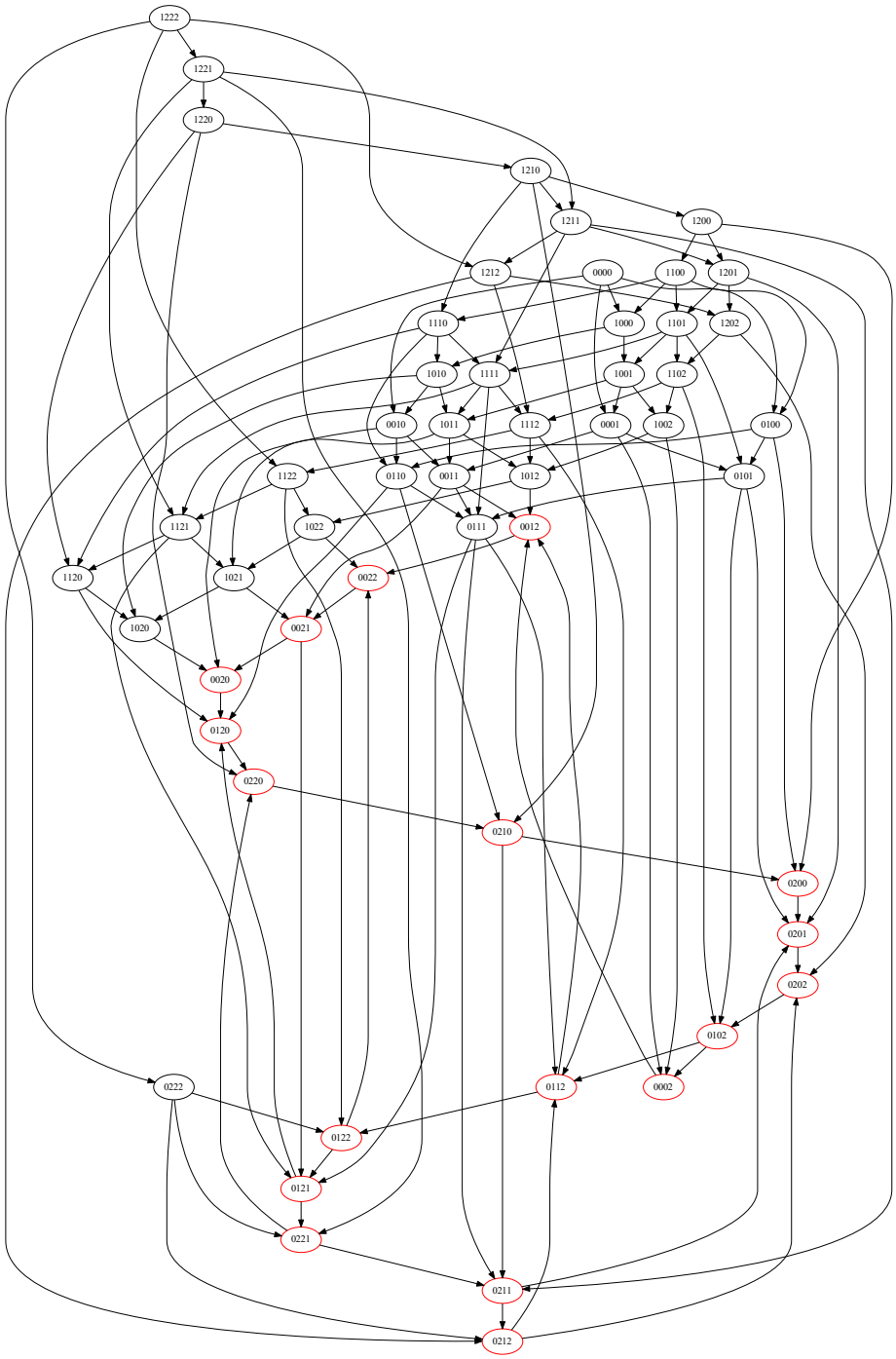


Figure 7.7 – Example of a discrete periodic attractor (red discrete states) that oscillates in 3 dimensions. The discrete model corresponds to the left influence graph in Fig 7.8, with $s(G_1 \to G_0) = s(G_2 \to G_0) = s(G_3 \to G_0) = 1$.

Figure 7.8 – The two influence graphs which have discrete periodic attractors oscillating in 3 or 4 dimensions. All arcs represent inhibitions between genes.

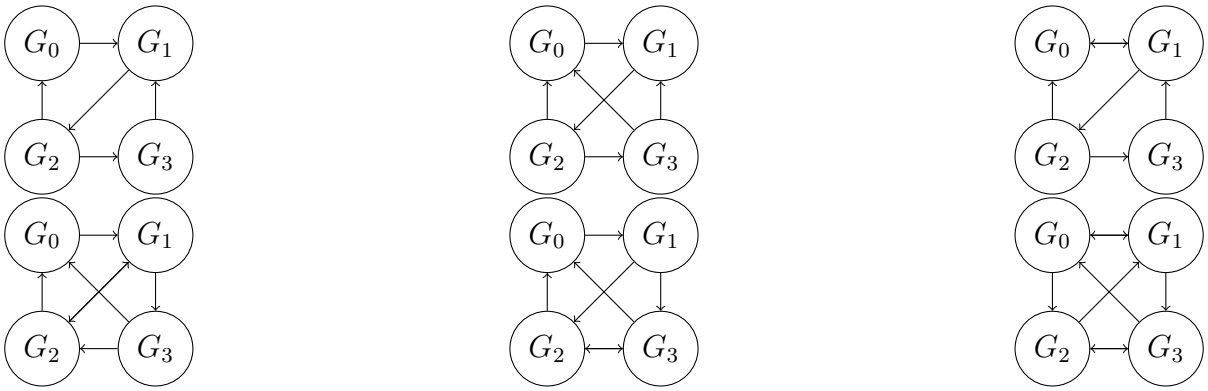


Figure 7.9 – The six influence graphs which have discrete periodic attractors oscillating only in 4 dimensions. All arcs represent inhibitions between genes.

the relations between thresholds and we find a sufficient and necessary condition with a simple form, describing the topology of the influence graph, for the existence of a discrete periodic attractor.

We use an exhaustive and computational approach to find this condition and we find some patterns in this condition. In our next step, we would like to prove this result in a more mathematical way, and try to extend this result for repressilators in N dimensions.

The topological feature used in this work, that is, the total out-degree of cycles of length n, could be potentially simplified based on other more common features. If it is possible, the simplified result could be more easily used for the design of new synthetic circuits.

Meanwhile, only the topology of influence graphs is considered in this work. For future works, we will also investigate how the relations between thresholds influence some complex dynamical properties.

Chapter 8

Conclusion and Perspectives

8.1 Summary of Contributions and Limits

With a lower dynamical complexity compared to ODEs and other hybrid systems, HGRN is a promising formalism to understand the global dynamical properties of gene regulatory networks. Methods related to HGRNs have not been exhaustively explored in the literature. In this thesis, we mainly focus on developing new analysis methods of HGRNs. There are two major research directions: The first one is to develop general analysis methods and the second one is to analyze some specific networks modeled as HGRNs.

For the general analysis methods, we propose a limit cycle analysis method which can identify all limit cycles inside simple cycles of discrete states (limit cycles that do not cross the same discrete state for more than once in one period) and analyze their stability; we also propose a reachability analysis method which can verify if the hybrid trajectory from a given singular hybrid state can reach certain region (a set of hybrid states). Among the state-of-the-art HGRNs of gene regulatory networks, these methods can fulfill successfully their tasks. While for general HGRNs (including other HGRNs which can model systems that are not gene regulatory networks), these methods might still have some limits.

For both methods, we ignore the existence of non-deterministic hybrid states. Indeed, for the current existing HGRNs of gene regulatory networks, ignoring these states does not influence the analysis results, mainly because such cases are very rare based on current observations. But in the future, we might have to consider non-deterministic behaviors. In that case, these algorithms should become more complex (for the identification of limit cycles, there will be more connections in the graph of discrete domains; for the reachability analysis method, a singular hybrid state could lead to several distinct hybrid trajectories), so some optimizations of these algorithms are required; and the method to analyze the stability of limit cycles should be adapted, because we might need to consider several Poincaré maps.

For the limit cycle analysis method, for now we only consider limit cycles inside simple cycles of discrete states, while, a priori, there could exist limit cycles that cross the same discrete state or the same discrete domain for more than once in one period. In that case, the enumeration of all possible cycles that could potentially contain limit cycles, becomes more difficult. In order to solve this problem, more mathematical tools are likely to be required.

For the reachability analysis method, for now we have only solved the reachability problem of hybrid trajectories with regular behaviors. Whether chaotic hybrid trajectories are decidable or not is still an open question. Therefore, exploring the boundary between decidable hybrid trajectories and undecidable ones should be an interesting theoretical study. There are also open questions related to the discrimination of chaotic hybrid trajectories: Whether we can find a sufficient and necessary condition for a hybrid trajectory to be chaotic.

In our works of analyzing some specific networks, we focus on a class of networks called repressilators. We first study the 3-dimensional repressilator using HGRNs and then study 4-dimensional repressilators using discrete models. There are also some improvements that can be done for these works.

The analysis of the 3-dimensional repressilator is based on the analysis of the stability of the characteristic state, which, in this case, has only one cycle of discrete states around it. For a HGRN of a 4-dimensional repressilator, a characteristic state can have several cycles of discrete states around it, so the application of Poincaré map becomes more complex. For now, we have not yet found a method to analyze such a characteristic state. In order to find such a method, we might need to consider other methods from control theory.

In the analysis of periodic attractors in discrete 4-dimensional repressilators, we find new topological features to describe a sufficient and necessary condition for the existence of a periodic attractor. We use an exhaustive strategy to prove that it is a sufficient and necessary condition because there is a finite and not too large number of influence graphs. If we could further prove it in a more mathematical way, then this result could be potentially extended for n-dimensional repressilators.

The final objective of these works on repressilators is to implement a synthetic repressilator with a controllable period into a living cell. Thanks to this thesis, some conditions about oscillation have been found. The next step is to compare these theoretical results with practice. So a continuation of this work is to collaborate with a team of biologists

to verify whether these conditions are compatible with the real systems.

We have also investigated the parameter identification methods based on time series data. An open question of this work is that whether we can ensure that the global minimum of the objective function is found, which could be an interesting continuation of this work.

8.2 Perspectives

Following the works of this thesis, there are other interesting directions that can be explored.

Predict Possible Long-term Continuous Behaviors

There is a strong link between HGRNs and discrete models: The transition graph of discrete states of a HGRN corresponds to the transition graph of discrete states of the associated discrete model following the asynchronous semantics. The major difference between these two frameworks is that the celerities of HGRNs give extra information about the long-term continuous behaviors. Indeed, in discrete models, for example, we can identify discrete periodic attractors, but we do not know whether the continuous oscillations in these discrete periodic attractors are sustained oscillations, damped oscillations or chaotic attractors. By sampling many HGRNs from a discrete model under certain hypothesis of the distribution of celerities and by using our methods to classify trajectories (introduced in Chapter 4), we could predict possible long-term continuous behaviors of a discrete model, for example, we can know if a system can have sustained oscillations or a stable fixed point, depending on the parameters. Furthermore, we can also predict the possible co-existence of attractors, for example, a co-existence of a stable limit cycle and a stable fixed point. Theoretically this approach can be applied on HGRNs in any dimension. But for models in higher dimensions, a large amount of simulations is required, which can be time consuming. To deal with this problem, it could be useful to explore new methods for the simulation of HGRNs.

Improve the Algorithm to Identify Discrete Attractors

In our work of 4-dimensional repressilators, we find a condition, which is described by topological features of the influence graphs, for the existence of a discrete periodic attractor. If we can extend these results for n-dimensional repressilators (the form of

the condition can be different in n-dimensional cases) and understand better how these topological features lead to periodic attractors, then these new results could be potentially used to accelerate the identification of attractors in more general discrete models. For instance, for certain discrete models, they could potentially give extra information about the existence or the location of certain periodic attractors.

Networks Inference based on HGRNs

In the parameter identification part of this thesis, we assume that we already know the influence graph. In some cases, such information is however unknown. It could be interesting to investigate inference methods of influence graphs based on HGRNs, which can lead to automatic construction of HGRNs with little biological knowledge (by combining the parameter identification approaches proposed in this thesis). In the literature, there are methods that can infer influence graphs based on discrete models [75, 99]. These methods can be potentially extended for HGRNs by not only considering the discrete levels, but also considering the signs of the time derivatives estimated from data.

Applications on Other Networks

Overall, some methods of this thesis have not yet been applied on specific biological problems. If some suitable applications can be found, it could be interesting to further investigate the merits of these proposed methods. Such applications can be, for example, to find all periodic trajectories of a given system, or to estimate the continuous basin of attraction of certain attractors. Also, even though this framework is proposed to model gene regulatory networks, it has potential to solve problems of other classes of networks, for example, ecological networks [100, 101], because the dynamics of HGRNs can be considered as a simplification of certain types of ordinary differential equations.

BIBLIOGRAPHY

1. Almeida, S., Chaves, M. & Delaunay, F., Control of synchronization ratios in clock/cell cycle coupling by growth factors and glucocorticoids, *Royal Society Open Science* **7,** 192054 (2020).
2. Barik, D., Baumann, W. T., Paul, M. R., Novak, B. & Tyson, J. J., A model of yeast cell-cycle regulation based on multisite phosphorylation, *Molecular systems biology* **6,** 405 (2010).
3. Karlebach, G. & Shamir, R., Modelling and analysis of gene regulatory networks, *Nature reviews Molecular cell biology* **9,** 770–780 (2008).
4. Kauffman, S. A., Metabolic stability and epigenesis in randomly constructed genetic nets, *Journal of theoretical biology* **22,** 437–467 (1969).
5. Thomas, R., Boolean formalization of genetic control circuits, *Journal of theoretical biology* **42,** 563–585 (1973).
6. Thomas, R., Regulatory networks seen as asynchronous automata: a logical description, *Journal of theoretical biology* **153,** 1–23 (1991).
7. Khalis, Z., Comet, J.-P., Richard, A. & Bernot, G., The SMBioNet method for discovering models of gene regulatory networks, *Genes, genomes and genomics* **3,** 15–22 (2009).
8. Bernot, G., Comet, J.-P. & Khalis, Z., *Gene regulatory networks with multiplexes in European simulation and modelling conference proceedings* (2008), 423–432.
9. Boyenval, D., Bernot, G., Collavizza, H. & Comet, J.-P., *What is a cell cycle checkpoint? the TotemBioNet answer in Computational Methods in Systems Biology: 18th International Conference, CMSB 2020, Konstanz, Germany, September 23–25, 2020, Proceedings 18* (2020), 362–372.
10. Behaegel, J., Comet, J.-P., Bernot, G., Cornillon, E. & Delaunay, F., A hybrid model of cell cycle in mammals, *Journal of bioinformatics and computational biology* **14,** 1640001 (2016).

11. Cornillon, E., Comet, J.-P., Bernot, G. & Enée, G., Hybrid gene networks: a new framework and a software environment, *advances in Systems and Synthetic Biology* (2016).

12. Comet, J.-P., Fromentin, J., Bernot, G. & Roux, O., *A formal model for gene regulatory networks with time delays in International Conference on Computational Systems-Biology and Bioinformatics* (2010), 1–13.

13. Sun, H., Folschette, M. & Magnin, M., *Limit cycle analysis of a class of hybrid gene regulatory networks in International Conference on Computational Methods in Systems Biology* (2022), 217–236.

14. Sun, H., Folschette, M. & Magnin, M., *Reachability Analysis of a Class of Hybrid Gene Regulatory Networks in Reachability Problems* (eds Bournez, O., Formenti, E. & Potapov, I.) (Springer Nature Switzerland, Cham, 2023), 56–69, ISBN: 978-3-031-45286-4.

15. Belgacem, I., Gouzé, J.-L. & Edwards, R., *Control of negative feedback loops in genetic networks in 2020 59th IEEE Conference on Decision and Control (CDC)* (2020), 5098–5105.

16. Firippi, E. & Chaves, M., Topology-induced dynamics in a network of synthetic oscillators with piecewise affine approximation, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **30,** 113128 (2020).

17. Mestl, T., Lemay, C. & Glass, L., Chaos in high-dimensional neural and gene networks, *Physica D: Nonlinear Phenomena* **98,** 33–52 (1996).

18. Edwards, R., Analysis of continuous-time switching networks, *Physica D: Nonlinear Phenomena* **146,** 165–199 (2000).

19. Edwards, R. & Glass, L., A calculus for relating the dynamics and structure of complex biological networks, *Adventures in Chemical Physics: A Special Volume of Advances in Chemical Physics* **132,** 151–178 (2005).

20. Farcot, E. & Gouzé, J.-L., Periodic solutions of piecewise affine gene network models with non uniform decay rates: the case of a negative feedback loop, *Acta Biotheoretica* **57,** 429–455 (2009).

21. Chaves, M. & Preto, M., Hierarchy of models: From qualitative to quantitative analysis of circadian rhythms in cyanobacteria, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **23,** 025113 (2013).

22. Maler, O. & Pnueli, A., *Reachability analysis of planar multi-linear systems in Computer Aided Verification: 5th International Conference, CAV'93 Elounda, Greece, June 28–July 1, 1993 Proceedings 5* (1993), 194–209.

23. Asarin, E., Maler, O. & Pnueli, A., Reachability analysis of dynamical systems having piecewise-constant derivatives, *Theoretical computer science* **138,** 35–65 (1995).

24. Sun, H., Comet, J., Folschette, M. & Magnin, M., *Condition for Sustained Oscillations in Repressilator Based on a Hybrid Modeling of Gene Regulatory Networks in Proceedings of the 16th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2023, Volume 3: BIOINFORMATICS, Lisbon, Portugal, February 16-18, 2023* (eds Ali, H., Deng, N., Fred, A. L. N. & Gamboa, H.) (SCITEPRESS, 2023), 29–40, https://doi.org/10.5220/0011614300003414.

25. Sun, H., Folschette, M. & Magnin, M., *Condition for Periodic Attractor in 4-Dimensional Repressilators in Computational Methods in Systems Biology* (eds Pang, J. & Niehren, J.) (Springer Nature Switzerland, Cham, 2023), 184–201, ISBN: 978-3-031-42697-1.

26. Chen, T., He, H. L. & Church, G. M., *in Biocomputing'99* 29–40 (World Scientific, 1999).

27. Goldbeter, A., Computational approaches to cellular rhythms, *Nature* **420,** 238–245 (2002).

28. Tyson, J. J. & Novak, B., Regulation of the eukaryotic cell cycle: molecular antagonism, hysteresis, and irreversible transitions, *Journal of theoretical biology* **210,** 249–263 (2001).

29. Ribeiro, T., Folschette, M., Magnin, M. & Inoue, K., Learning any semantics for dynamical systems represented by logic programs (2020).

30. Paulevé, L., Kolčák, J., Chatain, T. & Haar, S., Reconciling qualitative, abstract, and scalable modeling of biological networks, *Nature communications* **11,** 4256 (2020).

31. Gouzé, J.-L. & Sari, T., A class of piecewise linear differential equations arising in biological models, *Dynamical systems* **17,** 299–316 (2002).

32. Plahte, E. & Kjøglum, S., Analysis and generic properties of gene regulatory networks with graded response functions, *Physica D: Nonlinear Phenomena* **201,** 150–176 (2005).

33. Cornillon, E., *Modèles qualitatifs de réseaux génétiques: réduction de modèles et introduction d'un temps continu* PhD thesis (Université Côte d'Azur, 2017).

34. Abou-Jaoudé, W., Ouattara, D. A. & Kaufman, M., From structure to dynamics: frequency tuning in the p53–mdm2 network: I. logical approach, *Journal of theoretical biology* **258,** 561–577 (2009).

35. Alur, R., Henzinger, T. A., Lafferriere, G. & Pappas, G. J., Discrete abstractions of hybrid systems, *Proceedings of the IEEE* **88,** 971–984 (2000).

36. Alur, R., Dang, T. & Ivančić, F., *Counter-example guided predicate abstraction of hybrid systems in International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (2003), 208–223.

37. Znegui, W., Gritli, H. & Belghith, S., Design of an explicit expression of the Poincaré map for the passive dynamic walking of the compass-gait biped model, *Chaos, Solitons & Fractals* **130,** 109436 (2020).

38. Flieller, D., Riedinger, P. & Louis, J.-P., Computation and stability of limit cycles in hybrid systems, *Nonlinear Analysis: Theory, Methods & Applications* **64,** 352–367 (2006).

39. Girard, A., Computation and stability analysis of limit cycles in piecewise linear hybrid systems, *IFAC Proceedings Volumes* **36,** 181–186 (2003).

40. Hiskens, I. A., *Stability of hybrid system limit cycles: Application to the compass gait biped robot in Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)* **1** (2001), 774–779.

41. Geva-Zatorsky, N. *et al.,* Oscillations and variability in the p53 system, *Molecular systems biology* **2,** 2006–0033 (2006).

42. Batt, G., De Jong, H., Page, M. & Geiselmann, J., Symbolic reachability analysis of genetic regulatory networks using discrete abstractions, *Automatica* **44,** 982–989 (2008).

43. Xue, Y., Zhang, L. & Zhang, X., Reachable set estimation for genetic regulatory networks with time-varying delays and bounded disturbances, *Neurocomputing* **403,** 203–210 (2020).

44. Folschette, M., Paulevé, L., Magnin, M. & Roux, O., Sufficient conditions for reachability in automata networks with priorities, *Theoretical Computer Science* **608,** 66–83 (2015).

45. Paulevé, L., Reduction of qualitative models of biological networks for transient dynamics analysis, *IEEE/ACM transactions on computational biology and bioinformatics* **15,** 1167–1179 (2017).

46. Chai, X., Ribeiro, T., Magnin, M., Roux, O. & Inoue, K., Static analysis and stochastic search for reachability problem, *Electronic Notes in Theoretical Computer Science* **350,** 139–158 (2020).

47. Henzinger, T. A., Kopke, P. W., Puri, A. & Varaiya, P., *What's decidable about hybrid automata? in Proceedings of the twenty-seventh annual ACM symposium on Theory of computing* (1995), 373–382.

48. Asarin, E., Mysore, V. P., Pnueli, A. & Schneider, G., Low dimensional hybrid systems–decidable, undecidable, don't know, *Information and Computation* **211,** 138–159 (2012).

49. Frehse, G. *et al., SpaceEx: Scalable verification of hybrid systems in Computer Aided Verification: 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings 23* (2011), 379–395.

50. Dang, T. & Testylier, R., Reachability Analysis for Polynomial Dynamical Systems Using the Bernstein Expansion. *Reliab. Comput.* **17,** 128–152 (2012).

51. Sandler, A. & Tveretina, O., *Deciding Reachability for Piecewise Constant Derivative Systems on Orientable Manifolds in Reachability Problems: 13th International Conference, RP 2019, Brussels, Belgium, September 11–13, 2019, Proceedings 13* (2019), 178–192.

52. De Oliveira Oliveira, M. & Tveretina, O., *Mortality and Edge-to-Edge Reachability are Decidable on Surfaces in 25th ACM International Conference on Hybrid Systems: Computation and Control* (2022), 1–10.

53. Alur, R., Dang, T. & Ivančić, F., *Reachability analysis of hybrid systems via predicate abstraction in Hybrid Systems: Computation and Control: 5th International Workshop, HSCC 2002 Stanford, CA, USA, March 25–27, 2002 Proceedings 5* (2002), 35–48.

54. Asarin, E., Bournez, O., Dang, T. & Maler, O., *Approximate reachability analysis of piecewise-linear dynamical systems in Hybrid Systems: Computation and Control: Third International Workshop, HSCC 2000 Pittsburgh, PA, USA, March 23–25, 2000 Proceedings* (2002), 20–31.

55. Edwards, R & Glass, L, A calculus for relating the dynamics and structure of complex biological networks, *Advances in chemical physics* **132,** 151–178 (2006).

56. Hamatani, S. & Tsubone, T., Analysis of a 3-Dimensional Piecewise-Constant Chaos Generator Without Constraint, *IEICE Proceedings Series* **48** (2016).

57. Elowitz, M. B. & Leibler, S., A synthetic oscillatory network of transcriptional regulators, *Nature* **403,** 335–338 (2000).

58. Potvin-Trottier, L., Lord, N. D., Vinnicombe, G. & Paulsson, J., Synchronous long-term oscillations in a synthetic gene circuit, *Nature* **538,** 514–517 (2016).

59. Buşe, O., Pérez, R. & Kuznetsov, A., Dynamical properties of the repressilator model, *Physical Review E* **81,** 066206 (2010).

60. Dukarić, M. *et al.,* On three genetic repressilator topologies, *Reaction Kinetics, Mechanisms and Catalysis* **126,** 3–30 (2019).

61. Dilão, R., The regulation of gene expression in eukaryotes: bistability and oscillations in repressilator models, *Journal of theoretical biology* **340,** 199–208 (2014).

62. Kuznetsov, A & Afraimovich, V, Heteroclinic cycles in the repressilator model, *Chaos, Solitons & Fractals* **45,** 660–665 (2012).

63. Buşe, O., Kuznetsov, A. & Pérez, R. A., Existence of limit cycles in the repressilator equations, *International Journal of Bifurcation and Chaos* **19,** 4097–4106 (2009).

64. Müller, S. *et al.,* A generalized model of the repressilator, *Journal of mathematical biology* **53,** 905–937 (2006).

65. El Samad, H., Del Vecchio, D. & Khammash, M., *Repressilators and promotilators: Loop dynamics in synthetic gene networks in Proceedings of the 2005, American Control Conference, 2005.* (2005), 4405–4410.

66. Page, K. M. & Perez-Carrasco, R., Degradation rate uniformity determines success of oscillations in repressive feedback regulatory networks, *Journal of the Royal Society Interface* **15,** 20180157 (2018).

67. Wang, R., Chen, L. & Aihara, K., Construction of genetic oscillators with interlocked feedback networks, *Journal of theoretical biology* **242,** 454–463 (2006).

68. Dang, T. & Salinas, D., *Image computation for polynomial dynamical systems using the Bernstein expansion in International Conference on Computer Aided Verification* (2009), 219–232.

69. Garloff, J. & Smith, A. P., *A comparison of methods for the computation of affine lower bound functions for polynomials in International Workshop on Global Optimization and Constraint Satisfaction* (2003), 71–85.

70. Ziat, G., Maréchal, A., Pelleau, M., Miné, A. & Truchet, C., *Combination of Boxes and Polyhedra Abstractions for Constraint Solving in International Symposium on Formal Methods* (2019), 119–135.

71. Pelleau, M., Miné, A., Truchet, C. & Benhamou, F., *A constraint solver based on abstract domains in International Workshop on Verification, Model Checking, and Abstract Interpretation* (2013), 434–454.

72. Behaegel, J., Comet, J.-P. & Folschette, M., *Constraint identification using modified Hoare logic on hybrid models of gene networks in 24th International Symposium on Temporal Representation and Reasoning (TIME 2017)* **90** (2017).

73. Michelucci, R., Comet, J.-P. & Pallez, D., *Evolutionary continuous optimization of hybrid Gene Regulatory Networks in Biennial International Conference on Artificial Evolution (EA-2022)* (2022).

74. Byron, K. & Wang, J. T., A comparative review of recent bioinformatics tools for inferring gene regulatory networks using time-series expression data, *International journal of data mining and bioinformatics* **20,** 320–340 (2018).

75. Ribeiro, T., Folschette, M., Magnin, M. & Inoue, K., Learning any memory-less discrete semantics for dynamical systems represented by logic programs, *Machine Learning,* 1–78 (2021).

76. Burke, E. K., Burke, E. K., Kendall, G. & Kendall, G., *Search methodologies: introductory tutorials in optimization and decision support techniques* (Springer, 2014).

77. Liang, J. J., Qin, A. K., Suganthan, P. N. & Baskar, S, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE transactions on evolutionary computation* **10,** 281–295 (2006).

78. Liang, J. *et al.,* Classified perturbation mutation based particle swarm optimization algorithm for parameters extraction of photovoltaic models, *Energy Conversion and Management* **203,** 112138 (2020).

79. Eberhart, R. & Kennedy, J., *A new optimizer using particle swarm theory in MHS'95. Proceedings of the sixth international symposium on micro machine and human science* (1995), 39–43.

80. Kennedy, J. & Eberhart, R., *Particle swarm optimization in Proceedings of ICNN'95-international conference on neural networks* **4** (1995), 1942–1948.

81. Hansen, N., The CMA evolution strategy: a comparing review, *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms,* 75–102 (2006).

82. Storn, R. & Price, K., Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, *Journal of global optimization* **11,** 341 (1997).

83. Reeves, C. R., Genetic algorithms, *Handbook of metaheuristics,* 109–139 (2010).

84. Katoch, S., Chauhan, S. S. & Kumar, V., A review on genetic algorithm: past, present, and future, *Multimedia Tools and Applications* **80,** 8091–8126 (2021).

85. Such, F. P. *et al.,* Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning, *arXiv preprint arXiv:1712.06567* (2017).

86. Lim, S. M., Sultan, A. B. M., Sulaiman, M. N., Mustapha, A. & Leong, K. Y., Crossover and mutation operators of genetic algorithms, *International journal of machine learning and computing* **7,** 9–12 (2017).

87. Paulevé, L. & Richard, A., Static analysis of Boolean networks based on interaction graphs: a survey, *Electronic Notes in Theoretical Computer Science* **284,** 93–104 (2012).

88. Richard, A. & Comet, J.-P., Necessary conditions for multistationarity in discrete dynamical systems, *Discrete Applied Mathematics* **155,** 2403–2413 (2007).

89. Richard, A., Negative circuits and sustained oscillations in asynchronous automata networks, *Advances in Applied Mathematics* **44,** 378–392 (2010).

90. Remy, É., Ruet, P. & Thieffry, D., Graphic requirements for multistability and attractive cycles in a Boolean dynamical framework, *Advances in Applied Mathematics* **41,** 335–350 (2008).

91. Richard, A. & Tonello, E., Attractor separation and signed cycles in asynchronous Boolean networks, *Theoretical Computer Science,* 113706 (2023).

92. Page, K. M., Oscillations in well-mixed, deterministic feedback systems: Beyond ring oscillators, *Journal of Theoretical Biology* **481,** 44–53 (2019).

93. Perez-Carrasco, R. *et al.,* Combining a toggle switch and a repressilator within the AC-DC circuit generates distinct dynamical behaviors, *Cell systems* **6,** 521–530 (2018).

94. Goh, K.-I., Kahng, B. & Cho, K.-H., Sustained oscillations in extended genetic oscillatory systems, *Biophysical Journal* **94,** 4270–4276 (2008).

95. Tomazou, M., Barahona, M., Polizzi, K. M. & Stan, G.-B., Computational redesign of synthetic genetic oscillators for independent amplitude and frequency modulation, *Cell systems* **6,** 508–520 (2018).

96. Zhang, F. *et al.,* Independent control of amplitude and period in a synthetic oscillator circuit with modified repressilator, *Communications Biology* **5,** 23 (2022).

97. Melkman, A. A., Tamura, T. & Akutsu, T., Determining a singleton attractor of an AND/OR Boolean network in O (1.587 n) time, *Information Processing Letters* **110,** 565–569 (2010).

98. Akutsu, T., Kosub, S., Melkman, A. A. & Tamura, T., Finding a periodic attractor of a Boolean network, *IEEE/ACM transactions on computational biology and bioinformatics* **9,** 1410–1421 (2012).

99. Folschette, M. & Ribeiro, T., *GULA: Learning (From Any) Semantics of a Biological Regulatory Network in BIOSS–IA 2020 workshop* (2020).

100. Huang, Y. *et al.,* Matrix approach to land carbon cycle modeling: A case study with the Community Land Model, *Global change biology* **24,** 1394–1404 (2018).

101. Luo, M., Wang, S., Saavedra, S., Ebert, D. & Altermatt, F., Multispecies coexistence in fragmented landscapes, *Proceedings of the National Academy of Sciences* **119,** e2201503119 (2022).

COLLEGE ED MATHEMATIQUES
DOCTORAL STIC
PAYS DE LA LOIRE

CENTRALE NANTES

Titre : Identifier et analyser les comportements dynamiques à long terme des réseaux de régulation génétique à l'aide de modélisation hybride

Mot clés : Modélisation hybride, Cycle limite, Accessibilité, Répressilateur, Attracteur, Réseaux de régulation des gènes

Résumé : Utiliser des modèles dynamiques pour révéler les propriétés dynamiques des réseaux de régulation des gènes peut nous aider à mieux comprendre la nature de ces systèmes biologiques et à développer nouveaux traitements médicaux. Dans cette thèse, nous nous concentrons sur une classe de systèmes dynamiques hybrides appelés réseaux de régulation des gènes hybrides (HGRN) et visons à analyser les propriétés dynamiques à long terme. Nous proposons des méthodes pour trouver des cycles limites et analyser leur stabilité, et pour analyser l'accessibilité dans HGRNs. Ceci est suivi d'une étude plus approfondie de certains réseaux d'intérêt pour la biologie des systèmes : Les répressilateurs, et nous trouvons des conditions pour l'existence d'oscillations soutenues dans le répressilateur canonique en dimension 3, et des conditions, décrites par les caractéristiques topologiques des réseaux, pour l'existence d'un attracteur périodique dans les répressilateurs discrets en dimension 4. En résumé, cette thèse propose de nouvelles méthodes pour analyser certaines propriétés des HGRNs qui n'ont pas été étudiées auparavant, par exemple la stabilité des cycles limites à N dimensions, l'accessibilité, etc. Les résultats pourront être développés à l'avenir pour étudier d'autres grands réseaux complexes.

Title: Identifying and Analyzing Long-term Dynamical Behaviors of Gene Regulatory Networks with Hybrid Modeling

Keywords: Hybrid modeling, Limit cycle, Reachability, Repressilator, Attractor, Gene regulatory networks

Abstract: Using dynamical models to reveal dynamical properties of gene regulatory networks can help us better understand the nature of these biological systems and develop new medical treatments. In this thesis, we focus on a class of hybrid dynamical systems called Hybrid Gene Regulatory Network (HGRN) and aim to analyze long-term dynamical properties. We propose methods to find limit cycles and analyze their stability, and to analyze the reachability in HGRNs. This is followed by a deeper study of some networks of interest for Systems Biology: The repressilators, and we find conditions for the existence of sustained oscillations in the 3-dimensional canonical repressilator, and conditions, which are described by topological features of the networks, for the existence of a periodic attractor in discrete 4-dimensional repressilators. In summary, this thesis proposes new methods to analyze some properties of HGRNs that were not investigated before, for instance, the stability of N-dimensional limit cycles, the reachability, etc. The results can be further developed in the future to study other large complex networks.