UNIVERSITÀ DEGLI STUDI DI CAMERINO

International School of Advanced Studies

Computer Sciences and Mathematics

Stability Analysis and Control of Several Classes of Logical Dynamic Systems and the Applications in Game Theory

Doctor of Philosophy in
Computer Sciences and Mathematics
XXXIV cycle

Advisor
Prof. Renato De Leone

CoAdvisors:
Prof. Jianwei Xia, Prof. Lishan Qiao

Ph.D. Candidate:
Jianjun Wang

Academic Year 2019-2022

To my mom

Acknowledgements

It is a pleasure to thank many people who made this thesis possible.

Firstly, I would like to express my gratitude and appreciation to my advisor, Prof. Renato De Leone, for his patient guidance, constant encouragement, and the generous sharing of his immense specialized knowledge. He is the reason why I choose UNICAM to pursue my Ph.D. degree. I am deeply grateful to my co-advisors, Prof. Jianwei Xia and Prof. Lishan Qiao, for their continuous support in my research and greatly assistance in my life and career. I could not imagine better advisors for my doctoral study.

Importantly, a special thank to Dr. Shihua Fu, my unofficial co-supervisor, for having significantly contributed to my research. Hours and hours of discussions, suggestions and disagreements allowed me to understand the field and the nature of research itself. She is not just my instructor, but my sister and friend.

I owe my gratitude to Prof. Jianli Zhao for offering me the opportunity to pursue my doctoral degree, and his meticulous concern.

In addition, I wold like to express my sincere appreciation to Prof. Daizhan Cheng (the founder of STP) and Prof. Jun-e Feng, for discussing with me many of the technical details and providing me with selfless help. Their rigorous attitude towards academic research and their broad and deep knowledge have all affected me greatly.

I am very grateful to Dr. Cristina Soave, for assisting me in many different aspects. I am also very thankful to Elisa Marcelli, for sharing endless advice on study and life in Camerino.

I would like to thank my roommates: Hongliang Chen, Kemal Kilic and Ahmad Shafiei. With their openness and warmth, the time I spent in UNICAM is enjoyable and memorable. I would also like to thank the members of Research Center of STP of Matrices: Theory and Applications. In particular, many thanks to Yanan Pan, Chunfeng Jiang and Wen Liu. I cannot never forget their hospitality.

Finally, my gratitude goes to my parents, for their untiring support and trust. Last but not least, I have to thank my wife, Lanlan He, for her unconditional love and patience.

Abstract

With the rapid development of complex networks, logical dynamic systems have been commonly used mathematical models for simulating Genetic Regulatory Networks (GRNs) and Networked Evolutionary Games (NEGs), which have attracted considerable attention from biology, economy and many other fields. By resorting to the Semi-Tensor Product (STP) of matrices, logical dynamic systems can be equivalently converted into discrete time linear systems with algebraic forms. Based on that, this thesis analyzes the stability and studies the control design problems of several classes of logical dynamic systems. Moreover, the obtained results are applied to investigate the control and optimization problems of NEGs. The main results of this thesis are the following.

- The stability and event-triggered control for a class of k-Valued Logical Networks (KVLNs) with time delays are studied. First, some necessary and sufficient conditions are obtained to detect the stability of Delayed k-Valued Logical Networks (DKVLNs). Second, the global stabilization problem under event-triggered control is considered, and some necessary and sufficient conditions are presented for the stabilization of Delayed k-Valued Logical Control Networks (DKVLCNs). Moreover, an algorithm is proposed to construct all the event-triggered state feedback controllers via antecedence solution technique.

- The robust control invariance and robust set stabilization problems for a class of Mix-Valued Logical Control Networks (MVLCNs) with disturbances are studied. First, a calculation method for the Largest Robust Control Invariant Set (LRCIS) contained in a given set is introduced. Second, based on the Robust Control Invariant Subset (RCIS) obtained, the robust set stabilization of MVLCNs is discussed, and some new results are presented. Furthermore, the design algorithm of time-optimal state feedback stabilizers via antecedence solution technique is derived.

- The robust set stability and robust set stabilization problems for a class of Probabilistic Boolean Control Networks (PBCNs) with disturbances are studied. An algorithm to determine the Largest Robust Invariant Set (LRIS) with probability 1 of a given set for a Probabilistic Boolean Network (PBN) is proposed, and the necessary and sufficient conditions to detect whether the PBN is globally finite-time stable to this invariant set with probability 1 are established. Then, the PBNs with control inputs are considered, and an algorithm for LRCIS with probability 1 is provided, based on which, some necessary and sufficient conditions for finite-time robust set stabilization with probability 1 of PBCNs are presented. Furthermore, the design scheme of time-optimal state feedback stabilizers via antecedence solution technique is derived.

- The stabilization and set stabilization problems for a class of Switched Boolean Control Networks (SBCNs) with periodic switching signal are studied. First, algebraic forms are constructed for SBCNs with periodic switching signal. Second, based on the algebraic formulations, the stabilization and set stabilization of SBCNs with periodic switching signal are discussed, and some new results are presented. Furthermore, constructive procedure of open loop controllers is given, and the design algorithms of switching-signal-dependent state feedback controllers via antecedence solution technique are derived.

- The dynamics and control problems for a class of NEGs with time-invariant delay in strategies are studied. First, algebraic forms are constructed for Delayed Networked Evolutionary Games (DNEGs). Second, based on the algebraic formulations, some necessary and sufficient conditions for the global convergence of desired strategy profile under a state feedback event-triggered controller are presented. Furthermore, the constructive procedure and the number of all valid event-triggered state feedback controllers are derived, which can make the game converge globally.

- The evolutionary dynamics and optimization problems of the networked evolutionary boxed pig games with the mechanism of passive reward and punishment are studied. First, an algorithm is provided to construct the algebraic formulation for the dynamics of this kind of games. Then, the impact of reward and punishment parameters on the final cooperation level of the whole network is discussed.

Keywords: Semi-tensor product of matrices; Logical dynamic systems; Networked evolutionary games; Stability; Stabilization; Antecedence solution technique

Contents

List of Figures

List of Tables

List of Symbols

The next list describes the symbols that will be used within the thesis.

$\mathbb{R}$ The set of real numbers

$\mathbb{Z}_+$ The set of positive integers

$\mathbb{N}$ The set of natural numbers

$\mathbb{R}_{m\times n}$ The set of $m \times n$ real matrices

$[\alpha : \beta]$ The set of integers k satisfying $\alpha \leq k \leq \beta$

$\ltimes$ The semi-tensor product of matrices

$\otimes$ The Kronecker product of matrices

$*$ The Khatri-Rao product of matrices

I_k The k dimensional identify matrix

$[A]_{i,j}$ The (i, j)-element of matrix A

$\mathrm{Col}_i(A)$ The i-th column of matrix A

$\mathrm{Row}_i(A)$ The i-th row of matrix A

$\mathrm{Col}(A)$ The set of columns of matrix A

$\mathrm{Row}(A)$ The set of rows of matrix A

$\mathcal{D}_k$ $\{\frac{k-i}{k-1} \mid i \in [1 : k]\}$

$\mathcal{D}$ $\mathcal{D}_2$

δ_k^i $\mathrm{Col}_i(I_k)$

Δ_k $\{\delta_k^i \mid i \in [1 : k]\}$

Δ Δ_2

$\mathcal{D}_k^n$ $\underbrace{\mathcal{D}_k \times \cdots \times \mathcal{D}_k}_{n}$

Δ_k^n $\underbrace{\Delta_k \times \cdots \times \Delta_k}_{n}$

$\delta_m[i_1\ i_2\ \cdots\ i_n]$ $[\delta_m^{i_1}\ \delta_m^{i_2}\ \cdots\ \delta_m^{i_n}]$

$\mathfrak{B}_{m\times n}$ The set of $m \times n$ Boolean matrices
$\{A \in \mathbb{R}_{m\times n} \mid [A]_{i,j} = 0\ or\ 1\}$

$\mathbf{0}_{m\times n}$ A matrix with all entries equal to 0

$\mathbf{1}_{m\times n}$ A matrix with all entries equal to 1

$\mathcal{L}_{m\times n}$ The set of $m \times n$ logical matrices
$\{A \in \mathbb{R}_{m\times n} \mid \mathrm{Col}_i(A) \in \Delta_m, i \in [1:n]\}$

$lcm(n,p)$ The least common multiple of n and p

$|\cdot|$ The cardinal number of a set

$:=$ "defined as"

$A \le B$ $[A]_{i,j} \le [B]_{i,j}$

$A_{|W} \le B_{|W}$ $\mathrm{Col}_{i_j}(A) \le \mathrm{Col}_{i_j}(B)$, $\forall j = [1:\alpha]$, where $W = \{\delta_n^{i_1}, \delta_n^{i_2}, \cdots, \delta_n^{i_\alpha}\} \subseteq \Delta_n$

$V_r(A)$ Row stacking form of matrix A

Introduction

Logical Dynamic Systems

Inspired by the human genome project, a newly developing discipline called systems biology [45, 50], has been a widely focused research field, in which the essence of life phenomenon is well studied. More precisely, systems biology investigates the dynamic behavior and the interaction relationship of all the cells, proteins, DNAs and RNAs instead of the composed individual element of the biological system. The target of systems biology is to understand and express life systems from the systematic level, including the cellular networks and GRNs. In the early 1960s, Jacob and Monod found that "any cell contains a number of 'regulatory genes' that act as switches and can turn one another on and off" [117]. It shows that a genetic network is a logical one. Moreover, the logical nature of a cell network was also pointed out by Paul Nurse: "the cell machines then need to be linked and integrated together to define the modules and overall regulatory networks required to bring about the reproduction of the cell. This task will require system analyzes that emphasize the logical relationships between elements of the networks" [103]. Therefore, the logical dynamic systems have naturally been a powerful tool in describing, analyzing and simulating cellular networks or GRNs [39, 49, 110].

Logical network is a discrete-time nonlinear networked system, where all the state, input and output variables take finite values. When the gene state only takes logical values 1 or 0, that is, the gene expression is quantified to two different levels: active or inactive, the logical networks become Boolean Networks (BNs), which were firstly introduced by Kauffman [49] to model GRNs. In a BN, the evolution of each state variable relies on a pre-assigned logical function, which is determined by its neighbouring genes, itself and some logical operators. Furthermore, in order to describe the therapeutic interventions in GRNs, the concept of Boolean Control Networks (BCNs) was formally proposed in [45]. Due to their simple structure and parameter free, both BNs and BCNs have drawn a large amount of attention in biological systems [3]. Moreover, they have been used to simulate many biological models, such as segment polarity genes [3], λ phage[1] decision circuit [52]. The gene interactions of λ phage are summarized in Figure 1, where two genes, cI and cro directly affect this decision. When cI is ON (OFF) and cro is OFF (ON), the phage is in the lysogenic (lytic) state. Whether or not gene cI will be switched on, depends on a subtle control process, which is determined by interaction between five phage genes, cI, cro, cII, $cIII$, N, and the environmental state u.

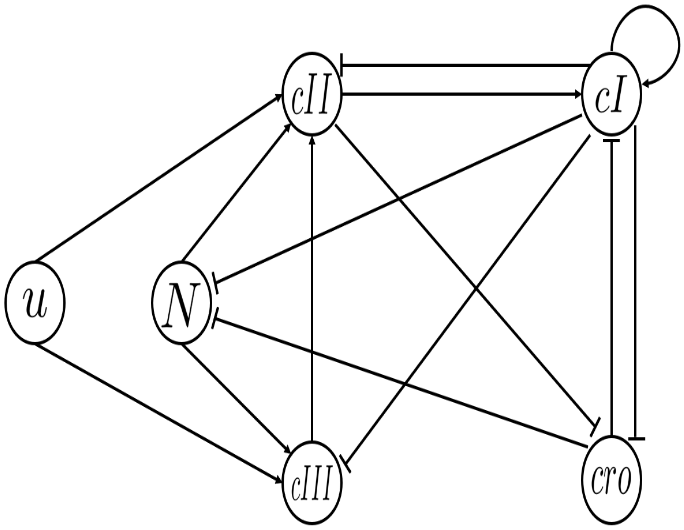

Figure 1: Gene interactions for the λ switch. The edges represents either activation ($\rightarrow$) or inhibition ($\dashv$).

However, many complex real-world networks can not be described by BNs. For example, in some biological models, the gene states are not limited to active or inactive, or when one gene is not strongly affected by another gene, binary variables will not be able to accurately describe the relationship between these genes. In a public goods game, each player may have a same number of investment schemes (more than 2), or more generally, each player may have a different number of investment schemes to compete with others. Based on that, the more general logical networks, namely KVLNs and Mixed-Valued Logical Networks (MVLNs) were proposed [20, 9]. Both of them have similar structure to BNs, but the values that can be assigned at the nodes are different from BNs. In a KVLN, all the states take value from a finite set $\mathcal{D}_k = \{\frac{k-i}{k-1} \mid i \in [1:k]\}$ and in an MVLN, each state takes value from a finite set of different size $\mathcal{D}_{k_i} = \{\frac{k_i-j}{k_i-1} \mid j \in [1:k_i]\}$, where $i \in [1:n]$. Thus, KVLN and MVLN can approximate a real cellular regulatory network better than a BN. As claimed in [2], "One of the major goals of systems biology is to develop a control

[1]The λ phage is a virus that grows on a bacterium.

theory for complex biological systems", the interests in k-Valued Logical Control Networks (KVLCNs) and MVLCNs have been increasing [51, 114]. Given a logical network, the basic interesting topics are to study its topological structure [4, 41], dynamic characteristics [1] and modeling and analysis of biological systems [43]. The existing research methods mainly include physical statistical method, graph theory method and computer simulation method [2, 42]. However, a unified theoretical framework could not be established to analyze the dynamic process of logical networks based on these methods. For example, only the fixed points and cycles for a specific system are computed in [4, 41], instead of providing a unified computation formula. Furthermore, since the dynamics of logical system are a process of logical evolution, and there are less tools for logical operations, investigating logical network becomes difficult. Hence, the investigation for logical dynamic system calls for a new tool.

STP of matrices was first introduced by Prof. Cheng to deal with Morgen's problem in 2001 [11]. It is a generalization of the conventional matrix product to the case that the dimension matching condition is not satisfied. STP almost keeps all the major properties of the conventional matrix product unchanged and has certain commutative properties, called pseudo-commutativity. Because of these advantages, the STP is capable of dealing with multi-linear and nonlinear functions. Using STP, a logical function can be converted into a multi-linear mapping, called the matrix expression of logic [14]. Under this transformation, an algebraic state- space representation approach can be established for logical dynamic systems. In the light of algebraic formulation and classical control theory, many major issues about the topology of logical networks including fixed points, cycles, basin of attractors, and transient times, can be revealed easily by a set of formulas [12]. Moreover, a multitude of fundamental and important results have been investigated for the analysis and control of logical dynamic systems based on the algebraic state-space representation approach, which include controllability [13, 17, 92, 150, 144, 145], observability [25, 132, 151, 35, 84, 149, 134], stability and stabilization [19, 58, 36, 62, 47, 66, 74, 154, 113, 86, 153, 93], disturbance decoupling [82, 10, 135, 129, 55, 119, 88], input-output decoupling [105, 118, 30], optimal control [16, 27, 143, 152, 125] and other related problems [72, 70, 24, 157]. It is worth noting that [13] won "Automatica Paper Prize-Theory & Method (2008-2010)" issued by IFAC in 2011, which shows that STP method is superior to other mathematical tools in dealing with logical dynamic systems. On the other hand, STP method has been applied to engineering related fields, such as power system [97], finite automata [126], information security [89] and vehicle control [124].

As we all know, time delay is a very common phenomenon and frequently occurs in real-world systems, such as transportation systems, chemical processes and communication systems. For instance, the information cannot be communicated instantaneously, and there may exists time delay due to some physical factors. The slow process of transcription, translocation and translation in GRNs may cause time delay. As a source of instability, time delays are unavoidable in various cases, and may result in poor performance. Moreover, the presence of time delays makes the analysis and control of logical networks much more difficult and challenging. Therefore, it is interesting and significant to investigate logical networks with time delays. Various works reported the theory and application of delayed logical networks based on STP method. For example, [57] and [38] investigated the controllability of BCNs with time-invariant delays in states and BCNs with time-invariant delays both in states and inputs respectively. Furthermore, necessary and sufficient conditions were presented to detect the controllability of BCNs. Compared with [57] and [38], a more practical time delay was encountered in [136], and the controllability and observability of BCNs with time-variant delays in states were studied. Moreover, the stability and stabilization problems of BNs with time delays were investigated in [85, 100]. It is worth noting that [100] first discussed the impact of stochastic delays on the dynamics of BNs.

Switching phenomenon between different models is widely observed in the real GRNs, which may be triggered by inherent mechanisms of systems, external disturbances or asynchronous behavior of GRNs. For instance, the growth and division of eukaryotic cells consist of four processes, which are activated by a set of discrete events [54]. The genetic switch in λ phage consists of two distinct models: lysis and lysogeny [115]. Compared with ordinary BNs, the existence of switched signals make the investigation of SBCNs much more complicated. These facts reveal that the study of SBCNs is a meaningful and challenging topic. Up to now, using STP method, many interesting results have been obtained for SBCNs [79, 131, 73, 137, 78]. The switching form considered in the above publications is an arbitrary transformation. However, in many practical biological systems, the switching behavior between different subsystems is not arbitrary, but usually relies on certain biological rhythms. For example, photosynthetic rate has a periodicity related to sunlight, the oscillation period of enzyme synthesis and enzyme activity is from one to dozens of minutes. Zou and Zhu [156] mentioned that the physical meaning of periodically time-variant BNs just lies in the periodic model transition among different BNs. Therefore, it seems that periodical switching signal is more suitable for SBCNs to stimulate biological cycle phenomenon. However, how the periodic switching signals affect the dynamic behavior of SBCNs has not been fully investigated.

Note that the deterministic rigidity of traditional BN (BCN) limits the further application in GRNs, since biological uncertainty and random perturbation always exist in real GRNs, and these phenomena can not be described via classical models. Hence, Shmulevich et al. [110] proposed the PBN model, which can be regarded as an undetermined system switching with a certain probability distribution among different sub-networks. Similarly, a PBN with exogenous control inputs is briefly called a PBCN. The main advantage of the PBN (PBCN) model over the deterministic BN (BCN) is that it cannot only share the appealing properties of BN (BCN) but also cope with the presence of random perturbation. PBN and PBCN have been recognized as the more flexible mathematical models of GRNs, and the theoretical and practical importance of probabilistic models have been shown in [37, 147, 57, 68, 75].

In a real GRN, external disturbances which maybe originate from gene mutations and recombination are ubiquitous [8]. These unavoidable disturbances may steer the system dynamics to some undesired behaviours [109]. For examples, cancer can be regarded as the failure of organism in resisting uncertainties including gene mutations. In some practical NEGs, the attackers can be regard as the disturbances to the strategy evolutionary dynamics of the games [112]. These cases show that it is indispensable to study the stability problem of logical networks with disturbances. There are some works concerning robust control invariance and robust set stabilization of BCN [67, 127, 71, 81, 83]. In particular, Li et al. presented necessary and sufficient conditions for the robust stabilization of BCNs and the constructive procedure for the controller [67]. Moreover, in [127] necessary and sufficient conditions were proposed to detect whether the BCNs with impulsive effects can robust stabilize to a given state set under a given state feedback control. In addition, the robust control invariance was studied in [71], and all possible state feedback gain matrices were characterized.

Stability and stabilization are two basic and important issues for logical dynamic systems and play a key role in some applications such as the explanation of some living phenomena and the therapeutic interventions of disease. Note that stability is an inherent attribute of systems, and it describes whether the network can converge to a certain desired state or a state set, which are called stability [19] and set stability [36], respectively. On the other hand, the ultimate goal of the GRNs is to design an efficient therapeutic strategy such that the organism can reach and maintain a desirable state. However, the system usually can not naturally evolve to the target state, thus external actions are necessary. This is the significance of investigating stabilization problem [74, 26]. In other cases, it is essential to study whether the system can be driven to a desirable subset of the state space instead of a single point, which is known as set stabilization [36]. In fact, there are many typical applications of set stabilization, such as synchronization [91], partial stabilization [111] and output tracking [72]. Furthermore, the control design is always one of the most interesting topics for the stabilization problems of logical control systems, and diverse design schemes of controllers have been presented, such as reachable set approach [74], pinning control technique [92], event-triggered control technique [55] and sampled-data control technique [87]. Different control strategies have their own unique advantages. For instance, the main advantage of pinning control is that the desired control objective can be achieved by controlling a small fraction of nodes [92], event-triggered control can effectively reduce the control execution times [55] and sampled-data control can effectively reduce the time of control updates [87]. Recently, a new method based on antecedence solution [53] has been proposed and it has been used to design the state feedback controls. For example, using the antecedence solution method, [48] investigated the stabilization problem of generic logical systems. The dynamics and control problems of singular BCN were studied in [122] by constructing the truth matrix of antecedence solution. Moreover, there are several works concerning the existence of antecedence solution based on STP. For instance, [107] presented necessary and sufficient conditions to detect the existence of antecedence solution. It is recognized that the main advantages of antecedence solution technique are (i) the clear one-step evolutionary dynamics are presented by constructing a series of truth matrices; (ii) the computations involved are very easy and straightforward; (iii) the algorithm can be easily implemented with the help of software tools such as Matlab.

From the above discussion, it is clear that logical dynamic systems are widely used to simulate and analyze various complex networks. Particularly, the evolutionary dynamics of NEGs are a logical process, thus many theories and results of logical dynamic systems can be applied to the investigation of NEGs directly.

Networked Evolutionary Games

NEGs have many applications in biology, economy, physics [28, 44, 102] and other areas [104, 148]. In an NEG, nodes and edges represent players and the interactions among players, respectively. The topological structure among players is not neglected and every player only interacts with his neighbours in the network. That coincides with many practical economic activities, where each person only plays games with relatives, friends or business partners. Limited by the bounded rationality of the players, each player updates his strategy according to certain strategy adjustment rule, which is affected by the local information of his neighbours.

The strategies of players can be expressed as the truth values of logical networks and the strategy updating rule can be interpreted as propositional logic formulas, based on which, the dynamic process of the game can be transformed into a logical dynamic system. Particularly, the strategy evolutionary dynamics of an NEG were firstly expressed as a k-valued logical dynamic network in [18], which provided a precise mathematical model for NEGs. Since then, some classical results obtained in logical dynamic system have been used to analyze, control and optimize NEGs based on STP method, and many excellent results have been proposed. For example, the results for the deterministic logical dynamic systems have been applied to the dynamic analysis and strategy optimization control of NEGs, such as the (group) strategy consensus problem was transformed into the (set) stabilization problem of KVLNs in [140, 31]. The optimization problem of NEGs [32, 139] was converted to the global convergence problem of a KVLCN. Besides, the results on robust control theory of logical dynamic systems with state constraints have been applied to the investigation of NEGs with attackers and forbidden profiles [77]. And the results on probabilistic logical dynamic system have been applied to the study of stochastic evolutionary games [22]. Last but not least, other comprehensive introduction about the applications of logical networks to NEGs were reported [15, 29, 33, 141, 98, 133].

Compared with traditional methods, the main advantages of logical dynamic systems lie in the overall manipulation of NEGs. Using the structural matrix and other information of the system, the game problems can be equivalently transformed into the calculation and analysis of the corresponding strategy transformation matrix, which can be easily solved based on the classical matrix theory.

Motivations

In the above, we introduced the background and research status of several kinds of logical dynamic systems, and their applications to NEGs. From that, it is clear that the research on the stability and stabilization of logical networks is of great theoretical significance and of practical worth. However, there are still some problems worthy of further study.

First, for the logical networks with time-invariant delay, the stability analysis and controller design problems have not been fully investigated. Existing works mainly concentrate on the delayed BNs, there are few results available on the stability and stabilization of KVLNs with time delays. Moreover, for the controller design, there are still much room to reduce the control costs. To the best of our knowledge, the event-triggered control and antecedence solution technique have not been introduced into the investigation of DKVLNs before. Furthermore, the interactions between players in NEGs can not take place instantaneously and their reactions can not be immediate, which will inevitably cause time delays in strategies. Thus, the applications of KVLNs with time delays to DNEGs need to be further explored.

Second, the robust (control) invariance of logical dynamic system has not been fully studied. The robust set stability (stabilization) means that all the initial states can converge to the robust (control) invariant set under the influence of disturbances. Thus, in order to solve the robust set stability (stabilization) problem, we should first study its robust (control) invariance problem. However, due to the complicated effects of disturbances variables on system dynamics, there are no results available on the computation of robust invariant set (RIS) and the determination of robust set stability. Moreover, the existing works just present the criteria to determine whether or not a given set is a RCIS, when the given set is not a RCIS, there are no works concentrate on this situation. Thus, finding an effective algorithm to calculate the RIS (RCIS) of logical dynamic systems is a problem to be further studied. Besides, the previous models are limited to BNs and KVLNs, the robust set stability and robust set stabilization problems of MVLNs and PBNs are not investigated in the necessary detail. Compared with BNs and KVLNs, the structures of MVLNs and PBNs are more complex, hence, the results in BNs and KVLNs can not be generalized to MVLNs and PBNs easily.

Third, the impact of periodic switching signals on the dynamic behavior of SBCNs has not been fully investigated. The existing works mainly concentrate on the stabilization and set stabilization of SBCNs under arbitrary switching signals. However, the stabilization and set stabilization problems of SBCNs with periodic switching signals have not been studied. Moreover, for the controller design, the condition of switching-signal-dependent controller is less conservative than the one of the switching-signal-independent controller. Thus, how to design switching-signal-dependent state feedback controllers for the stabilization and set stabilization problems of periodic SBCNs need to be studied carefully.

Finally, the scheme to avoid the free-rider phenomenon in the networked evolutionary boxed pig games has not been fully investigated. Due to the lack of effective mathematical tools, it is hard to systematically analyze the influence of passive reward and punishment on the final cooperation level of the whole network. Thus, the investigation of the evolutionary dynamics and optimization problems of the boxed pig games with passive reward and punishment need to be further considered.

Main Contents

In reaction to the above problems, this thesis investigates the stability analysis and control of several types of logical dynamic systems and the applications in game theory. The main contents are summarized as follows:

Chapter 1 presents the preliminaries of this thesis, and mainly introduces the concept of STP and some basic properties, the algebraic state-space representation of logical dynamic systems, the related concepts of NEGs and its algebraic state-space expression, and the antecedence solution technique, which lay a theoretical foundation for the research of subsequent chapters.

Chapter 2 investigates the stability and event-triggered feedback control problems of DKVLNs. First, we provide the algebraic formulations of DKVLNs and DKVLCNs under the event-triggered control. Then, we present some necessary and sufficient conditions for the solvability of stability problem of DKVLNs. Moreover, we derive the necessary and sufficient conditions for the solvability of stabilization problem of DKVLCNs under the event-triggered control and establish an algorithm to design all the event-triggered state feedback stabilizers based on antecedence solution technique.

Chapter 3 investigates the robust control invariance and robust set stabilization problems of MVLCNs, and proposes a novel method to compute the LRCIS of MVLCNs and obtains all the possible state feedback controllers to keep the robust control invariance. We further present some necessary and sufficient conditions for the solvability of robust set stabilization of MVLCNs and provide an algorithm to design all the time-optimal state feedback controls.

Chapter 4 investigates the robust set stability and set stabilization problems of PBCNs. First, we introduce the concepts of RIS and RCIS with probability 1 of PBNs, respectively, and propose the algorithms to compute the LRIS and the LRCIS with probability 1, respectively. Second, we determine all the state feedback controls to keep the robust control invariance with probability 1 of PBCN. Third, we propose the concept of finite-time robust set stability of PBNs and provide the necessary and sufficient conditions for the solvability of finite-time robust set stability with probability 1. Fourth, we introduce the concept of finite-time robust set stabilization of PBCNs, and present the necessary and sufficient conditions for the solvability of finite-time robust set stabilization with probability 1. Moreover, we construct an algorithm to design all the time-optimal state feedback stabilizers for finite-time robust set stabilization of PBCNs.

Chapter 5 investigates the stabilization and set stabilization problems of periodic SBCNs. First, we introduce the model of periodic SBCNs, the concepts of stabilization, set stabilization, common control fixed point and common control invariant set of periodic SBCNs. Second, we give the necessary and sufficient conditions for the solvability of stabilization of periodic SBCNs, and present a constructive procedure of open loop controller and a design algorithms of switching-signal-dependent state feedback controller via antecedence solution technique. Furthermore, we drive the necessary and sufficient conditions for the solvability of set stabilization of periodic SBCNs, and provide an algorithm to construct all the switching-signal-dependent state feedback controllers.

Chapter 6 investigates the event-triggered control design problem for NEGs with time-invariant delay in strategies. First, we formulate the model of DNEGs under the Myopic Best Response Adjustment Rule (MBRAR), and present the algebraic expression of the dynamics of DNEGs. Then, we propose the necessary and sufficient conditions to detect whether the evolutionary dynamics can globally converge to the desired strategy profile. We further establish an algorithm to construct the event-triggered control to assure the global convergence of the game.

Chapter 7 investigates the algebraization and optimization problems of networked evolutionary boxed pig games with passive reward and punishment. First, we introduce the model of the boxed pig game with the mechanism of passive reward and punishment, and propose an algorithm to construct the algebraic formulation of the evolutionary dynamics under the Unconditional Imitation Strategy Updating Rule (UISUR). Then, we analyze the evolutionary dynamics of the game, and present the necessary and sufficient conditions for the global convergence to the full cooperation profile. Moreover, we discuss the impact of reward and punishment parameters on the final cooperation level.

Chapter 8 summarizes the results obtained in this thesis and points out the further research problems.

The main contents of this thesis are shown in Figure 2.

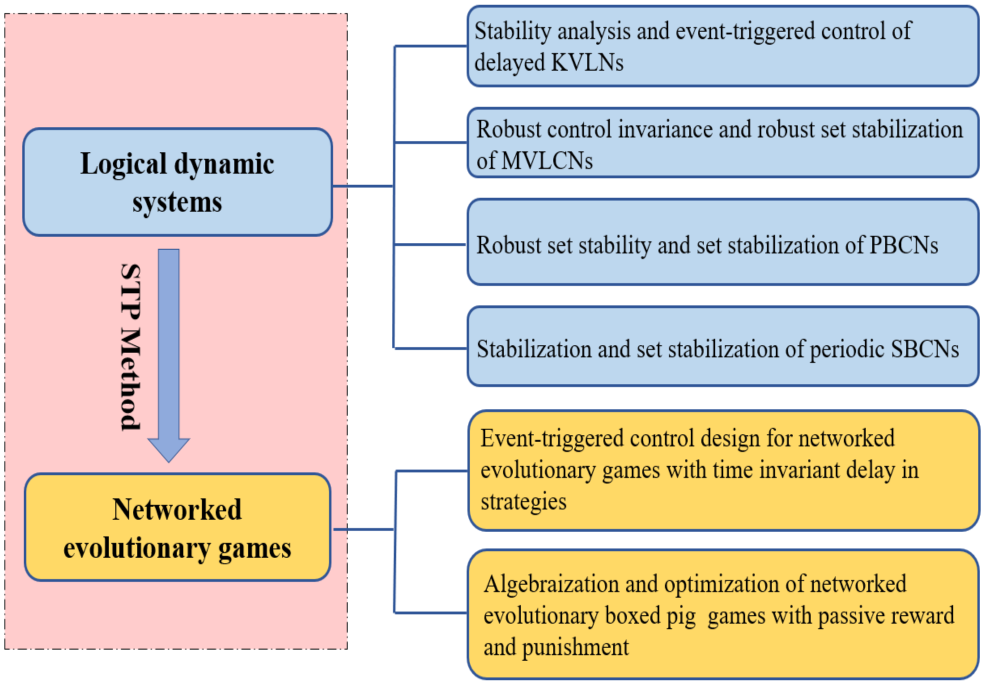

Figure 2: The structure of the thesis

Chapter 1

Preliminaries

In this chapter, we recall the definition and properties of STP, algebraic expression of mapping on finite set, some basic concepts and the matrix expression of NEGs and the antecedence solution technique, which will be used throughout this thesis.

1.1 Definition and properties of STP

This section mainly introduces the definition of STP and some basic properties related to this thesis. For more information on STP of matrices, please refer to [14].

Definition 1.1.1. *[14] The STP of two matrices $A \in \mathbb{R}_{m\times n}$ and $B \in \mathbb{R}_{p\times t}$ is defined as*

$$A \ltimes B = (A \otimes I_{\frac{\alpha}{n}})(B \otimes I_{\frac{\alpha}{p}}), \tag{1.1}$$

where $\alpha = lcm(n, p)$.

Remark 1.1.1. *Note that $A \ltimes B = AB$ when $n = p$. Therefore, the STP is a generalization of the ordinary matrix product and keeps the major properties of ordinary matrix product, such as associative law and distributive law. In this paper, the symbol "$\ltimes$" will be omitted without confusion.*

Proposition 1.1.1. *[14] The STP of matrices has the following properties:*

1. *(Associative law) Let $A \in \mathbb{R}_{m\times n}$, $B \in \mathbb{R}_{p\times q}$, $C \in \mathbb{R}_{r\times s}$, then*

$$(A \ltimes B) \ltimes C = A \ltimes (B \ltimes C).$$

2. *(Distributive law) Let $A, B \in \mathbb{R}_{m\times n}$, $C \in \mathbb{R}_{r\times s}$, then*

$$(A + B) \ltimes C = A \ltimes C + B \ltimes C,$$
$$C \ltimes (A + B) = C \ltimes A + C \ltimes B.$$

3. *(Pseudo-commutative law) Let* $X \in \mathbb{R}_{t\times 1}$*,* $A \in \mathbb{R}_{m\times n}$*, then*

$$X \ltimes A = (I_t \otimes A) \ltimes X. \tag{1.2}$$

Lemma 1.1.2. *[14] (Power-reducing matrix, Swap matrix and Front/Rear-maintaining operator)*

1. *Let* $x \in \Delta_n$*, then*

$$x \ltimes x = \Psi_n \ltimes x, \tag{1.3}$$

where $\Psi_n \in \mathcal{L}_{n^2\times n}$ *is called a power-reducing matrix, which is defined as*

$$\Psi_n = [\delta_n^1 \ltimes \delta_n^1 \quad \delta_n^2 \ltimes \delta_n^2 \quad \cdots \quad \delta_n^n \ltimes \delta_n^n].$$

2. *Let* $x \in \Delta_n$*,* $u \in \Delta_m$*, then*

$$x \ltimes u = W_{[m,n]} \ltimes u \ltimes x, \tag{1.4}$$

where $W_{[m,n]} \in \mathcal{L}_{mn\times mn}$ *is the so-called swap matrix, which is defined as*

$$W_{[m,n]} = [\delta_n^1 \ltimes \delta_m^1 \cdots \delta_n^n \ltimes \delta_m^1 \cdots \delta_n^1 \ltimes \delta_m^m \cdots \delta_n^n \ltimes \delta_m^m].$$

3. *Given two integers* $p \geq 2$*,* $q \geq 2$*. The "front-maintaining operator" and "rear-maintaining operator" are defined as*

$$D_f^{p,q} = I_p \otimes 1_q^T, \quad D_r^{p,q} = 1_p^T \otimes I_q. \tag{1.5}$$

Then

$$D_f^{p,q} xy = x, \qquad D_r^{p,q} xy = y,$$

where $x \in \Delta_p$*,* $y \in \Delta_q$*.*

1.2 Algebraic state-space representation of logical dynamic systems

Definition 1.2.1. *[14] A function* $f : \mathcal{D}_{k_1} \times \mathcal{D}_{k_2} \times \cdots \times \mathcal{D}_{k_n} \longrightarrow \mathcal{D}_{k_0}$ *is called a mix-valued logical function. If* $k_0 = k_1 = \cdots = k_n = k$*, then* f *is called a* k*-valued logical function. Particularly, when* $k = 2$*, it is called a Boolean function.*

For $\frac{k_i-j}{k_i-1} \in \mathcal{D}_{k_i}$, $j \in [1 : k_i]$, $i \in [1 : n]$, identify $\frac{k_i-j}{k_i-1}$ as a vector form $\delta_{k_i}^j$, then $\mathcal{D}_{k_i} \sim \Delta_{k_i}$. Under the vector form, the mix-valued logical function

$$f : \mathcal{D}_{k_1} \times \mathcal{D}_{k_2} \times \cdots \times \mathcal{D}_{k_n} \longrightarrow \mathcal{D}_{k_0}$$

can be equivalently expressed as

$$f : \Delta_{k_1} \times \Delta_{k_2} \times \cdots \times \Delta_{k_n} \longrightarrow \Delta_{k_0}.$$

Lemma 1.2.1. *[14] Let $f : \Delta_{k_1} \times \Delta_{k_2} \times \cdots \times \Delta_{k_n} \longrightarrow \Delta_{k_0}$ be a mix-valued logical function. Then there exists a unique matrix $M_f \in \mathcal{L}_{k_0 \times k}$, called the structural matrix of f, such that*

$$f(x_1, x_2, \cdots, x_n) = M_f \ltimes_{i=1}^{n} x_i, \tag{1.6}$$

where $\ltimes_{i=1}^{n} x_i = x_1 \ltimes x_2 \ltimes \cdots \ltimes x_n \in \Delta_k$, $k = \prod_{i=1}^{n} k_i$ and $x_i \in \Delta_{k_i}$.

Following, some frequently-used logical operators and their structural matrices are introduced.

- **Boolean function**

– Negation($\neg$): $\neg x = 1 - x$. Structural matrix: $M_n = \delta_2[2\ 1]$.

– Conjunction($\wedge$): $x \wedge y = \min\{x, y\}$. Structural matrix: $M_c = \delta_2[1\ 2\ 2\ 2]$.

– Disjunction($\vee$): $x \vee y = \max\{x, y\}$. Structural matrix: $M_d = \delta_2[1\ 1\ 1\ 2]$.

– Conditional($\rightarrow$): $x \rightarrow y = \neg x \wedge y$. Structural matrix: $M_i = \delta_2[1\ 2\ 1\ 1]$.

– Biconditional($\leftrightarrow$): $x \leftrightarrow y = (x \rightarrow y) \wedge (y \rightarrow x)$. Structural matrix: $M_e = \delta_2[1\ 2\ 2\ 1]$.

Example 1.2.1. *Assume*

$$f(x, y, z) = (x \wedge y) \vee (y \wedge \neg z),$$

where $x, y, z \in \mathcal{D}$.

Based on the vector form of logical variables and Lemma 1.2.1, we have

$$\begin{aligned}
f(x, y, z) &= (x \wedge y) \vee (y \wedge \neg z) \\
&= M_d (x \wedge y)(y \wedge \neg z) \\
&= M_d (M_c x y)(M_c y M_n z) \\
&= M_d M_c (I_4 \otimes M_c) x y^2 M_n z \\
&= M_d M_c (I_4 \otimes M_c) x \Psi_2 y M_n z \\
&= M_d M_c (I_4 \otimes M_c)(I_2 \otimes \Psi_2)(I_4 \otimes M_n) x y z \\
&= \delta_2[1\ 1\ 2\ 2\ 2\ 1\ 2\ 2] x y z.
\end{aligned}$$

- **k-valued logical function**

  In this case, the logical variables take value from $\mathcal{D}_k = \{\frac{k-i}{k-1} \mid i \in [1 : k]\}$. As an example, we take $k = 3$.

– Negation($\neg$): $\neg x = 1 - x$. Structural matrix: $M_{n,3} = \delta_3[3\ 2\ 1]$;

– Conjunction($\wedge$): $x \wedge y = \min\{x, y\}$. Structural matrix: $M_{c,3} = \delta_3[1\ 2\ 3\ 2\ 2\ 3\ 3\ 3\ 3]$.

– Disjunction($\vee$): $x \vee y = \max\{x, y\}$. Structural matrix: $M_{d,3} = \delta_3[1\ 1\ 1\ 1\ 2\ 2\ 1\ 2\ 3]$.

– Conditional($\rightarrow$): $x \rightarrow y = \neg x \wedge y$. Structural matrix: $M_{i,3} = \delta_3[1\ 2\ 3\ 1\ 2\ 2\ 1\ 1\ 1]$.

– Biconditional($\leftrightarrow$): $x \leftrightarrow y = (x \rightarrow y) \wedge (y \rightarrow x)$. Structural matrix: $M_{e,3} = \delta_3[1\ 2\ 3\ 2\ 2\ 2\ 3\ 2\ 1]$.

- **Mix-valued logical function**

Definition 1.2.2. *[14] Let $x \in \mathcal{D}_p$, $\phi_{[q,p]} : \mathcal{D}_p \longrightarrow \mathcal{D}_q$. ϕ is a projection of $\phi_{[q,p]}(x) := \xi \in \mathcal{D}_q$ and*

$$| \xi - x | = \min_{y \in \mathcal{D}_q} | x - y | .$$

Remark 1.2.1. *If there are two such solutions as $\xi_1 > x$ and $\xi_2 < x$, $\phi_{[q,p]}(x) := \xi_1$ is called the up-round projection and $\phi_{[q,p]}(x) := \xi_2$ is called the down-round projection. In the sequel, the default projection is the up-round projection. Moreover, denote the structural matrix of $\phi_{[q,p]}$ as $\Phi_{[q,p]}$. For example, $\Phi_{[3,2]} = \delta_3[1\ 3]$, $\Phi_{[2,3]} = \delta_2[1\ 2\ 2]$.*

Definition 1.2.3. *[14] Let σ be an unary operator on $\mathcal{D}_k$, and $x \in \mathcal{D}_p$. Then*

$$\sigma(x) := \sigma(\phi_{[k,p]}(x)) \in \mathcal{D}_k.$$

Let σ be a binary operator on $\mathcal{D}_k$, and $x \in \mathcal{D}_p$, $y \in \mathcal{D}_q$. Then

$$x\sigma y := (\phi_{[k,p]}(x))\sigma(\phi_{[k,q]}(y)) \in \mathcal{D}_k.$$

Example 1.2.2. *Consider logical function*

$$y = f(x_1, x_2, x_3) = x_1 \wedge (x_2 \leftrightarrow x_3),$$

where $x_1, x_3 \in \mathcal{D}$, $x_2, y \in \mathcal{D}_3$.

Its algebraic expression can be computed as

$$\begin{aligned} y = f(x_1, x_2, x_3) &= x_1 \wedge (x_2 \leftrightarrow x_3) \\ &= M_{c,3}(\Phi_{[3,2]}x_1)(M_{e,3}x_2(\Phi_{[3,2]}x_3)) \\ &= M_{c,3}\Phi_{[3,2]}(I_2 \otimes M_{e,3})(I_4 \otimes \Phi_{[3,2]})x_1x_2x_3 \\ &= \delta_3[1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 3\ 3\ 2\ 1]x_1x_2x_3. \end{aligned}$$

Definition 1.2.4. *[14] Let $M \in \mathbb{R}_{p\times m}$ and $N \in \mathbb{R}_{q\times m}$. Then, the Khatri-Rao product of M and N is defined as*

$$M * N = [\mathrm{Col}_1(M) \ltimes \mathrm{Col}_1(N), \cdots, \mathrm{Col}_m(M) \ltimes \mathrm{Col}_m(N)] \in \mathbb{R}_{pq\times m}.$$

Assume that a logical network with algebraic formulation is presented as follows

$$F : \begin{cases} y_1 = M_1 \ltimes_{i=1}^{n} x_i, \\ y_2 = M_2 \ltimes_{i=1}^{n} x_i, \\ \quad\vdots \\ y_m = M_m \ltimes_{i=1}^{n} x_i. \end{cases} \tag{1.7}$$

Then, with $x = \ltimes_{i=1}^{n} x_i$ and $y = \ltimes_{i=1}^{m} y_i$, we have

$$y = M_F x, \tag{1.8}$$

where $M_F = M_1 * M_2 * \cdots * M_m$.

In the following, an illustrative example is given to show how to convert the logical form of a BCN into the algebraic formulation.

Example 1.2.3. *A simple BCN model for the λ phage decision circuit system shown in Figure 1 can be derived as follows*

$$\begin{cases} x_1(t+1) = (\neg x_2(t)) \wedge (\neg x_5(t)), \\ x_2(t+1) = (\neg x_5(t)) \wedge (x_2(t) \vee x_3(t)), \\ x_3(t+1) = (\neg x_2(t)) \wedge u(t) \wedge (x_1(t) \vee x_4(t)), \\ x_4(t+1) = (\neg x_2(t)) \wedge u(t) \wedge x_1(t), \\ x_5(t+1) = (\neg x_2(t)) \wedge (\neg x_3(t)), \end{cases} \tag{1.9}$$

where state variables x_1, x_2, x_3, x_4 and x_5 represent genes N, cI, cII, $cIII$ and cro respectively, and control variable u denotes an external factor.

Based on the above discussion, the first expression becomes

$$\begin{aligned} x_1(t+1) &= (\neg x_2(t)) \wedge (\neg x_5(t)) \\ &= M_c M_n x_2(t) M_n x_5(t) \\ &= M_c M_n (I_2 \otimes M_n) x_2(t) x_5(t) \\ &= M_c M_n (I_2 \otimes M_n) D_r^{2^2,2} D_f^{2^3,2^2} u(t) x_1(t) x_2(t) x_3(t) x_4(t) x_5(t) \\ &:= M_1 u(t) x(t), \end{aligned}$$

where $x(t) = \ltimes_{i=1}^{5} x_i(t)$ and

$$\begin{aligned} M_1 \;=\; \delta_2[&2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1 \\ &2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1 \end{aligned}$$

$$\begin{array}{cccccccccccccccc} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1]. \end{array}$$

Similarly, we can obtain

$$x_i(t+1) := M_i u(t)x(t), \quad i \in [2:5],$$

where

$$\begin{aligned} M_2 &= \delta_2[2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 2\ 2\ 2 \\ &\quad 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 2\ 2\ 2 \\ &\quad 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 2\ 2\ 2 \\ &\quad 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 2\ 2\ 2], \\ M_3 &= \delta_2[2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ &\quad 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2 \\ &\quad 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2 \\ &\quad 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2], \\ M_4 &= \delta_2[2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ &\quad 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2 \\ &\quad 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2 \\ &\quad 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2], \\ M_5 &= \delta_2[2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1 \\ &\quad 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1 \\ &\quad 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1 \\ &\quad 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1]. \end{aligned}$$

Thus, the algebraic expression of system (1.9) can be easily computed as

$$x(t+1) = Lu(t)x(t), \tag{1.10}$$

where $L = M_1 * M_2 * M_3 * M_4 * M_5$ and

$$\begin{aligned} L &= \delta_{32}[32\ 24\ 32\ 24\ 32\ 24\ 32\ 24\ 26\ 2\ 26\ 2\ 25\ \ 9\ \ 25\ \ 9 \\ &\quad 32\ 24\ 32\ 24\ 32\ 24\ 32\ 24\ 28\ 4\ 32\ 8\ 27\ 11\ 31\ 15 \\ &\quad 32\ 24\ 32\ 24\ 32\ 24\ 32\ 24\ 32\ 8\ 32\ 8\ 31\ 15\ 31\ 15 \\ &\quad 32\ 24\ 32\ 24\ 32\ 24\ 32\ 24\ 32\ 8\ 32\ 8\ 31\ 15\ 31\ 15]. \end{aligned}$$

Remark 1.2.2. *The algebraic formulation of logical networks can be equivalently converted into a logical form, one can refer [14] for details. Moreover, a Matlab toolbox*[1] *has been developed to compute STP and convert the logical form and algebraic expression to each other, all examples in this thesis are computed based on this toolbox.*

The pseudo-logical function plays a key role in the investigation of NEGs. Here we introduce the concept and the algebraic expression of pseudo-logical function.

Definition 1.2.5. *[14] A function $f : \prod_{i=1}^{n} \mathcal{D}_{k_i} \longrightarrow \mathbb{R}$ is called a mix-valued pseudo-logical function. If $k_0 = k_1 = \cdots = k_n = k$, then it is called a k-valued pseudo-logical function. Particularly, when $k = 2$, it is called a pseudo-Boolean function.*

Under the vector form, the pseudo-logical function can be converted into algebraic expression.

Lemma 1.2.2. *[14] Let $f : \prod_{i=1}^{n} \Delta_{k_i} \longrightarrow \mathbb{R}$ be a mix-valued pseudo-logical function. Then, there exists a unique row vector $V_f \in \mathbb{R}_{1\times k}$, called the structural vector of f, such that*

$$f(x_1, x_2, \cdots, x_n) = V_f \ltimes_{i=1}^{n} x_i, \tag{1.11}$$

where $x_i \in \Delta_{k_i}$ and $k = \prod_{i=1}^{n} k_i$.

1.3 Algebraic state-space expression of NEGs

Definition 1.3.1. *[18] A normal finite game is a triplet (N, S, P), where*

(i) $N = \{1, 2, \cdots, n\}$ is the set of players;

(ii) for each player $i \in [1 : n]$ a strategy set $S_i = \{1, 2, \cdots, k_i\}$, $i \in [1 : n]$, is defined and $S := \prod\limits_{i=1}^{n} S_i$ is the set of profiles;

(iii) for each player $i \in [1 : n]$ a payoff function $p_i : S \to R$, $i \in [1 : n]$, P is the set of payoff functions.

Definition 1.3.2. *[18] A normal game with two players is called a Fundamental Network Game (FNG), if*

$$S_1 = S_2 := S_0 = \{1, 2, \cdots, k\}.$$

An FNG is symmetric, if $p_1(x, y) = p_2(y, x)$, $\forall x, y \in S_0$, where $p_i = p_i(x, y)$ is the payoff function of player i, $i = 1, 2$.

[1] *http://lsc.amss.ac.cn/∼dcheng/stp/STP.zip*

Definition 1.3.3. *[18] An NEG is a triplet* $((N, \mathcal{E}), G, \Pi)$*, where*

(i) $(N, \mathcal{E})$ *is a network graph, with* $N = \{1, 2, \cdots, n\}$ *the set of vertices,* $\mathcal{E} \subset N \times N$ *the set of edges. In the network, nodes and edges denote, respectively, players and interaction relationship among players.*

(ii) G *is an* FNG*, such that if* $(i, j) \in \mathcal{E}$*, then* i *and* j *play the* FNG *with strategies* $x_i(t)$ *and* $x_j(t)$*, respectively. FNG determines the type of the game.*

(iii) Π *is a local information based strategy updating rule.*

Definition 1.3.4. *[18] Let* N *be the set of nodes in network and* $\mathcal{E} \subset N \times N$ *be the set of edges:*

(i) if $(i, j) \in \mathcal{E}$ *implies* $(j, i) \in \mathcal{E}$*, the graph is undirected, otherwise, it is directed;*

(ii) $j \in N$ *is called a neighborhood node of* i*, if either* $(i, j) \in \mathcal{E}$ *or* $(j, i) \in \mathcal{E}$*. The set of neighborhood nodes of* i *is called the neighborhood of* i*, denoted by* $U(i)$*. In particular,* $i \in U(i)$*;*

(iii) ignoring the directions of edges, if there exists a path from i *to* j *with length less than or equal to* r*, then* j *is said to be an* r*-neighborhood node of* i*. The set of* r*-neighborhood node of* i *is denoted by* $U_r(i)$*.*

Definition 1.3.5. *[18] Let* $N = Z \cup U$ *be a partition of* N*. If the strategies of any* $u \in U$ *can be assigned arbitrarily, we call* $[(Z \cup U, E), G, \Pi]$ *a controlled NEGs. Moreover,* $u \in U$ *is called a control player and* $z \in Z$ *is called a state player.*

Next, an example is given to explain the definition of NEGs and neighbor of player, and show the effectiveness of STP method in converting the dynamics of the game into the corresponding algebraic formulation.

Example 1.3.1. *Consider an NEG with the following items:*

- $N = \{1,\ 2,\ 3,\ 4,\ 5,\ 6\}$ *is the player set. The network topological structure among players is shown in Figure 1.1.*

- *the FNG is the Prisoner's dilemma. The payoff bi-matrix is shown in Table 1.1, where* $T = 0,\ R = -1,\ P = -6,\ S = -9$*. If* $(i, j) \in \mathcal{E}$*, then players* i *and* j *can play the Prisoner's dilemma. For instance, player* 1 *plays game with players* 2 *and* 6*, however, he does not play games with other players.*

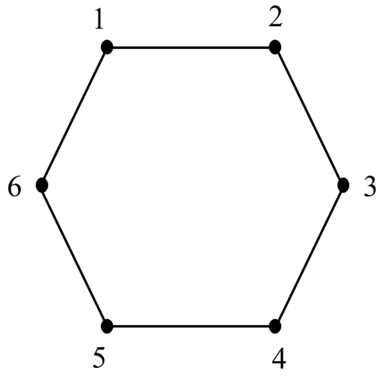

Figure 1.1: The network graph with six players

Table 1.1: Payoff bi-matrix of Prisoner's dilemma

$P_1 \setminus P_2$	C	D
C	$(R,\ R)$	$(S,\ T)$
D	$(T,\ S)$	$(P,\ P)$

- *the strategy updating rule is the MBRAR [23], that is, each player forecasts that his rivals will repeat their previous decisions, and the strategy choice at present time is the best response against his neighbors' strategies. Based on this, we have*

$$x_i(t+1) \in Q_i := \arg\max_{x_i \in \{C,D\}} p_i(x_i, x_j(t) \mid j \in U(i)\backslash\{i\}), \tag{1.12}$$

where $x_i(t)$ represents the strategy of player i at time t, $U(i)\backslash i$ means to remove i from $U(i)$. When the best response of player i is not unique, that is, $\mid Q_i \mid > 1$, then $x_i(t+1) = C$.

From Definition 1.3.4 and Figure 1.1, $U(1) = \{1,\ 2,\ 6\}$, $U_2(1) = \{1,\ 2,\ 3,\ 5,\ 6\}$, $U_3(1) = \{1, 2, 3, 4, 5, 6\}, U(1)\backslash\{1\} = \{2, 6\}, U(6)\backslash\{6\} = \{1, 5\}, U(i)\backslash\{i\} = \{i-1,\ i+1\}$, $i \in [2:5]$. Hence, the dynamics of the game can be expressed as follows

$$\begin{aligned} x_1(t+1) &= f_1(x_2(t), x_6(t)), \\ x_i(t+1) &= f_i(x_{i-1}(t), x_{i+1}(t)),\ i \in [2:5], \\ x_6(t+1) &= f_6(x_1(t), x_5(t)). \end{aligned}$$

Then, we need to compute the structural matrix of f_j, $j \in [1:6]$. Consider player 1 as an example. First, the dynamics of player 1 can be converted into the following algebraic formulation

$$x_1(t+1) = Fx_2(t)x_6(t).$$

From the parameters in Table 1.1, we conclude that for player 1, no matter which strategy his opponent chooses, he will choose defection at next time. Identify $C \sim \delta_2^1$, $D \sim \delta_2^2$. Then

$$x_1(t+1) = \delta_2^2, \quad \forall x_2(t), x_6(t) \in \{C, D\}.$$

Since there are four profiles for players 2 and 6, that is, both players 2 and 6 choose C, player 2 chooses C and player 6 chooses D, player 2 chooses D and player 6 chooses C, both players 2 and 6 choose D. Hence, the matrix F has four columns, where each column corresponds to the possible choice of player 1 at next time with four different profiles. For example, when players 2 and 6 choose C, we have $x_2(t)x_6(t) = \delta_2^1 \ltimes \delta_2^1 = \delta_4^1$. Thus,

$$x_1(t+1) = Fx_2(t)x_6(t) = F\delta_4^1 = \mathrm{Col}_1(F) = \delta_2^2,$$

from which we obtain the first column of structure matrix F. Similarly, we can compute the other columns of F:

$$\begin{aligned}\mathrm{Col}_2(F) &= F\delta_2^1 \ltimes \delta_2^2 = F\delta_4^2 = \delta_2^2,\\ \mathrm{Col}_3(F) &= F\delta_2^2 \ltimes \delta_2^1 = F\delta_4^3 = \delta_2^2,\\ \mathrm{Col}_4(F) &= F\delta_2^2 \ltimes \delta_2^2 = F\delta_4^4 = \delta_2^2.\end{aligned}$$

Therefore, $F = \delta_2[2\ 2\ 2\ 2]$.

From Lemma 1.1.2,

$$\begin{aligned}x_1(t+1) &= Fx_2(t)x_6(t)\\ &= FD_r^{2,2}x_1(t)x_2(t)x_6(t)\\ &= FD_r^{2,2}D_f^{2^2,2^3}x(t)\\ &:= F_1x(t),\end{aligned} \tag{1.13}$$

where $x(t) = \ltimes_{i=1}^6 x_i(t)$ and $F_1 = \delta_2[2\ 2\ \cdots\ 2] \in \mathcal{L}_{2\times 2^6}$.

Similarly, we can obtain the algebraic expression of f_i

$$x_i(t+1) = F_ix(t), \quad i \in [2:6], \tag{1.14}$$

where $F_i = \delta_2[2\ 2\ \cdots\ 2] \in \mathcal{L}_{2\times 2^6}$.

Multiplying the left and right of (1.13) and (1.14) by STP, we have

$$x(t+1) = Lx(t),$$

where $L = F_1 * F_2 * \cdots * F_6 = \delta_{2^6}[2^6\ 2^6\ \cdots\ 2^6] \in \mathcal{L}_{2^6\times 2^6}$. The matrix L is called the profile transition matrix of the game. Based on this matrix, we can calculate the final evolutionary dynamics of the game from any initial profile.

1.4 Antecedence solution technique

Definition 1.4.1. *[48] Let $X_1, X_2, \cdots, X_n, U_1, U_2, \cdots, U_m$ be Boolean variables. Given a set of Boolean equations*

$$F_i(X_1, X_2, \cdots, X_n, U_1, U_2, \cdots, U_m) = C_i, \quad i \in [1:s], \tag{1.15}$$

where $C_i \in \mathcal{D}$ is a constant. A set of Boolean functions

$$U_j = G_j(X_1, X_2, \cdots, X_n), \quad j \in [1:m], \tag{1.16}$$

is called an antecedence solution of (1.15), if (1.16) implies (1.15).

Let x_i, u_j and c_l be the vector form of X_i, U_j and C_l, respectively. Based on Lemma 1.2.1, the Boolean functions (1.15) and (1.16) can be expressed as follows:

$$M_F ux = c, \tag{1.17}$$

$$u = Gx, \tag{1.18}$$

where $x = \ltimes_{i=1}^{n} x_i \in \Delta_{2^n}$, $u = \ltimes_{j=1}^{m} u_j \in \Delta_{2^m}$, $c = \ltimes_{l=1}^{s} c_l \in \Delta_{2^s}$, $M_F \in \mathcal{L}_{2^s \times 2^{n+m}}$ and $G \in \mathcal{L}_{2^m \times 2^n}$.

The matrix $T \in \mathfrak{B}_{2^m \times 2^n}$ given by

$$[T]_{i,j} = \begin{cases} 1, & \text{if } M_F \delta_{2^m}^i \delta_{2^n}^j = c, \\ 0, & \text{otherwise,} \end{cases} \tag{1.19}$$

is called the truth matrix of (1.17).

The following lemma shows under which conditions the antecedence solution condition holds.

Lemma 1.4.1. *[48] The equation $u = Gx$ is an antecedence solution of $M_F ux = c$, if and only if $G \le T$, where T is given by (1.19).*

Replacing the single state $c \in \Delta_{2^s}$ in (1.17) by a set $\Omega \subset \Delta_{2^s}$ and choosing x from set W, concept of the generalized antecedence solution can be introduced.

Definition 1.4.2. *[48] Let $W \subset \Delta_{2^n}$ be a restricted set. Then (1.18) is called a W-antecedence solution of (1.17), if when $x \in W$ and $u = Gx$ then $M_F ux \in \Omega$ holds.*

Similarly, the truth matrix $T_{\Omega|W} \in \mathfrak{B}_{2^m \times 2^n}$ of (1.17) with respect to Ω and restricted on W can be constructed as follows:

$$[T_{\Omega|W}]_{i,j} = \begin{cases} 1, & \text{if } M_F \delta_{2^m}^i \delta_{2^n}^j \in \Omega, \quad \forall \delta_{2^n}^j \in W, \\ 0, & \text{otherwise.} \end{cases} \tag{1.20}$$

The below lemma shows under which conditions the generalized antecedence solution condition holds.

Lemma 1.4.2. *[48] The equation $u = Gx$ is a W-antecedence solution of $M_F ux \in \Omega$, if and only if $G_{|W} \leq T_{\Omega|W}$, where $T_{\Omega|W}$ is given by (1.20) and $x \in W$.*

In the following, an example is given to show how to verify the antecedence solution and the generalized antecedence solution.

Example 1.4.1. *Consider*

$$\begin{aligned} F_1(X_1, X_2, U) &= (X_1 \vee U) \to X_2 = 1, \\ F_2(X_1, X_2, U) &= X_1 \wedge U = 0. \end{aligned} \tag{1.21}$$

Let x_1, x_2 and u be the vector form of X_1, X_2 and U respectively, and identify $1 \sim \delta_2^1$ and $0 \sim \delta_2^2$. Based on Lemma 1.2.1, the algebraic formulation of (1.21) can be easily calculated as follows

$$M_f ux = \delta_4^2, \tag{1.22}$$

where $x = x_1 \ltimes x_2$ and

$$M_f = \delta_4[1\ 3\ 2\ 4\ 2\ 4\ 2\ 2].$$

The truth matrix of (1.21) is given by

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}. \tag{1.23}$$

Note that since there is no $G \in \mathcal{L}_{2\times 4}$ such that $G \leq T$, thus, (1.21) has no antecedence solution.

Let $W = \{\delta_4^1, \delta_4^3, \delta_4^4\}$, replacing state δ_4^2 in (1.22) by $\Omega = \{\delta_4^2, \delta_4^4\}$, the truth matrix is given by

$$T_{\Omega|W} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}. \tag{1.24}$$

Hence, from Lemma 1.4.2, there exists a W-antecedence solution of $M_F ux \in \Omega$ and all the solutions can be obtained as

$$\begin{aligned} u &= \delta_2[2\ 1\ 1\ 1]x, \\ u &= \delta_2[2\ 1\ 1\ 2]x, \\ u &= \delta_2[2\ 1\ 2\ 1]x, \\ u &= \delta_2[2\ 1\ 2\ 2]x. \\ u &= \delta_2[2\ 2\ 1\ 1]x, \\ u &= \delta_2[2\ 2\ 1\ 2]x, \\ u &= \delta_2[2\ 2\ 2\ 1]x, \\ u &= \delta_2[2\ 2\ 2\ 2]x. \end{aligned}$$

Chapter 2

Stability Analysis and Event-triggered Control of Delayed k-Valued Logical Networks

2.1 Introduction

A KVLN is a discrete-time nonlinear networked system, where all the state, input and output variables take value from a finite set [6]. It has captured wide attention of numerous scholars from different areas, including gene regulation [2, 49, 123], combinational logic circuit design [90], NEG [18], finite automata [126], information security [89] and so on. Using STP method, one can easily convert the dynamics of KVLNs into an equivalent algebraic form. Up to now, many significant results of KVLN have been obtained via STP method, ranging among controllability [60], output tracking [64], optimal control [143], and other problems [80, 63, 21].

As we all know, time delay is very common in real-world system, such as transportation systems, chemical processes and communication systems. For example, time delay is associated with the slow process of transcription, translation, and translocation or the finite switching speed of amplifiers in GRNs [146, 85]. As a source of instability, time delay phenomenon is unavoidable in various cases and may result in some poor performance. Moreover, the presence of time delay causes difficulties and challenges in the stability analysis and control design. Therefore, it is interesting and significant to investigate KVLN with time delays. The existing works mainly concentrate on delayed BNs with very few results on the stability and stabilization of DKVLNs.

Event-triggered control is an effective control strategy and has been broadly used to study logical control networks since it was first introduced into the investigation of disturbance decoupling problem of BNs [55]. This control scheme consists of two elements: a feedback controller that determines the control input, and a triggering mechanism that decides when the controller need to be updated again [40]. The main advantage of event-triggered control is that the control execution times and the computation costs can be greatly reduced. Thus, the event-triggered state feedback control is utilized to study the stabilization of DKVLNs.

This chapter investigates the stability and event-triggered control design problems of DKVLNs via the truth matrices technique. The main contributions are:

- Necessary and sufficient conditions for the stability of DKVLNs are established.
- Necessary and sufficient conditions for the stabilization of DKVLCNs under the event-triggered control are given.
- A design procedure to compute all the event-triggered state feedback controllers is presented.

2.2 Problem formulation

The dynamics of KVLNs with state delay can be described as follows:

$$\begin{cases} X_1(t+1) = f_1(X(t-\tau+1)), \\ X_2(t+1) = f_2(X(t-\tau+1)), \\ \qquad\qquad \vdots \\ X_n(t+1) = f_n(X(t-\tau+1)), \end{cases} \tag{2.1}$$

where $\tau \in \mathbb{Z}_+$ denotes the time delay, $f_i : \mathcal{D}_k^n \to \mathcal{D}_k$, $i \in [1:n]$ are k-valued logical functions and $X(t) = (X_1(t), X_2(t), \cdots, X_n(t)) \in \mathcal{D}_k^n$ are states at time t.

Let x_i be the vector form of logical variable X_i, $i \in [1:n]$. Then, based on Lemma 1.2.1, system (2.1) can be converted into

$$x(t+1) = Lx(t-\tau+1), \tag{2.2}$$

where $L \in \mathcal{L}_{k^n \times k^n}$ and $x(t) = \ltimes_{i=1}^n x_i(t) \in \Delta_{k^n}$.

Similarly, a DKVLCN with n nodes, m control inputs is described as follows:

$$\begin{cases} X_1(t+1) = g_1(X(t-\tau+1), U(t-\tau+1)), \\ X_2(t+1) = g_2(X(t-\tau+1), U(t-\tau+1)), \\ \qquad\qquad \vdots \\ X_n(t+1) = g_n(X(t-\tau+1), U(t-\tau+1)), \end{cases} \tag{2.3}$$

where $g_i : \mathcal{D}_k^{m+n} \to \mathcal{D}_k$, $i \in [1:n]$ are k-valued logical functions and $U(t) = (U_1(t), U_2(t), \cdots, U_m(t)) \in \mathcal{D}_k^m$ are control inputs at time t.

Let u_j be the vector form of U_j, again, following Lemma 1.2.1, system (2.3) can be expressed as

$$x(t+1) = \widetilde{L}u(t-\tau+1)x(t-\tau+1), \tag{2.4}$$

where $\widetilde{L} \in \mathcal{L}_{k^n \times k^{m+n}}$ and $u(t) = \ltimes_{j=1}^m u_j(t) \in \Delta_{k^m}$.

The event-triggered control mechanism is an intermittent control scheme based on a triggering event set $\Gamma \subseteq \Delta_{k^n}$. When the current state does not belong to the set Γ, no control is activated and the dynamics of the system will evolve desirably in the form (2.2). Otherwise, the control input is operated and the system will maintain in the form (2.4). Therefore, the overall dynamics of DKVLCNs under the event-triggered control can be expressed as

$$x(t+1) = \begin{cases} Lx(t-\tau+1), & x(t-\tau+1) \in \Delta_{k^n} \backslash \Gamma, \\ \widetilde{L}u(t-\tau+1)x(t-\tau+1), & x(t-\tau+1) \in \Gamma. \end{cases} \tag{2.5}$$

Equivalently, the algebraic form of DKVLCNs with event-triggered control can be reformulated as

$$x(t+1) = [\widetilde{L}\ L]\overline{u}(t-\tau+1)x(t-\tau+1) := \overline{L}\overline{u}(t-\tau+1)x(t-\tau+1), \tag{2.6}$$

where $\overline{L} \in \mathcal{L}_{k^n \times k^n(k^m+1)}$, and the event-triggered control $\overline{u}(t-\tau+1) \in \Delta_{k^m+1}$ is constructed from $u(t-\tau+1)$ as follows:

$$\overline{u}(t-\tau+1) = \begin{cases} \delta_{k^m+1}^{k^m+1}, & x(t-\tau+1) \in \Delta_{k^n} \backslash \Gamma, \\ [u(t-\tau+1)^T\ 0]^T, & x(t-\tau+1) \in \Gamma, \end{cases} \tag{2.7}$$

where $\delta_{k^m+1}^{k^m+1}$ denotes no control action.

In the next sections, the dynamics of DKVLNs (2.2) will be analyzed and all the event-triggered state feedback controls

$$\overline{u}(t) = Hx(t), \tag{2.8}$$

where $H \in \mathcal{L}_{(k^m+1) \times k^n}$, are designed, such that DKVLCNs (2.5) are globally stabilizable.

2.3 Stability analysis of DKVLNs

Definition 2.3.1. *DKVLN (2.2) is said to be globally stable to a given state $x^* \in \Delta_{k^n}$, if for any initial states $x(-\tau+1), x(-\tau+2), \cdots, x(0) \in \Delta_{k^n}$, there exists an integer $T^* \geq 0$, such that $x(t) = x^*$, $\forall t \geq T^*$.*

Note that if DKVLN (2.2) is globally stable to $x^* = \delta_{k^n}^{\alpha} \in \Delta_{k^n}$, then x^* is a fixed point of the system. However, the reverse does not hold. In the following, we discuss how to determine whether system (2.2) is globally stable to $x^* = \delta_{k^n}^{\alpha}$.

For any $\rho \in \mathbb{Z}_+$ and any initial state $x(t_0) \in \{x(-\tau+1), x(-\tau+2), \cdots, x(0)\}$, the dynamics of system (2.2) can be expressed as

$$
\begin{aligned}
x(t_0+\tau) &= Lx(t_0), \\
x(t_0+2\tau) &= Lx(t_0+\tau) \\
&= L^2 x(t_0), \\
&\vdots \\
x(t_0+\rho\tau) &= Lx(t_0+(\rho-1)\tau) \\
&= L^2 x(t_0+(\rho-2)\tau) \\
&\vdots \\
&= L^{\rho} x(t_0).
\end{aligned}
$$

Now, we are ready to present the following necessary and sufficient conditions for the global stability of DKVLNs (2.2).

Theorem 2.3.1. *Consider system (2.2) with initial states $x(-\tau+1), x(-\tau+2), \cdots, x(0)$, and the given objective state $x^* = \delta_{k^n}^{\alpha}$. Then system (2.2) is globally stable to $x^* = \delta_{k^n}^{\alpha}$, if and only if there exists an integer $T^* \in [1 : k^n - 1]$, such that*

$$
\mathrm{Col}(L^{T^*}) = \{\delta_{k^n}^{\alpha}\}. \tag{2.9}
$$

Remark 2.3.1. *The equation (2.9) is easy to be verified via Matlab. However, when the dimension of the structural matrix L is large, the iteration process will take a lot of time according to Theorem 2.3.1. Next, we try to find another way to detect the global stability of system (2.2).*

First, let $R_0 = \{\delta_{k^n}^{\alpha}\}$ and construct the truth matrix $T_{R_0} \in \mathfrak{B}_{k^n \times k^n}$:

$$
[T_{R_0}]_{i,j} = \begin{cases} 1, & \text{if } i = \alpha, [L]_{\alpha,j} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.10}
$$

Compute $R_1 = \{\delta_{k^n}^j \mid \mathrm{Col}_j(T_{R_0}) \neq \mathbf{0}_{k^n}\}$. Then, for $t \in \mathbb{Z}_+$, construct a series of truth matrices $T_{R_t} \in \mathfrak{B}_{k^n \times k^n}$ as follows

$$[T_{R_t}]_{i,j} = \begin{cases} 1, \text{ if } L_{i,j} = 1, \forall \delta_{k^n}^i \in R_t, \\ 0, \text{ otherwise}, \end{cases} \tag{2.11}$$

and compute $R_{t+1} = \{\delta_{k^n}^j \mid \mathrm{Col}_j(T_{R_t}) \neq \mathbf{0}_{k^n}\}$. In fact, from the construction of T_{R_t}, $t \in \mathbb{N}$, we have the following three cases:

Case 1: if $[T_{R_0}]_{\alpha,\alpha} = 1$, then $T_{R_{t-1}} \leq T_{R_t}$, for all $t \geq 1$, or equivalently, $R_t \subseteq R_{t+1}$.

Case 2: if $[T_{R_0}]_{\alpha,\alpha} = 1$ and $[T_{R_0}]_{\alpha,j} = 0$, $\forall j \neq \alpha$, then $[T_{R_{t-1}}]_{\alpha,\alpha} = 1$ and $[T_{R_{t-1}}]_{\alpha,j} = 0$, $\forall j \neq \alpha$, for all $t \geq 1$, or equivalently, $R_t = R_0$.

Case 3: if $T_{R_{\lambda-1}} = T_{R_\lambda}$, for some $\lambda \geq 1$, then $T_{R_{t-1}} = T_{R_{\lambda-1}}$, $\forall t \geq \lambda$, or equivalently, $R_t = R_\lambda$.

The following criteria are proposed to detect the global stability of DKVLNs (2.2).

Theorem 2.3.2. *Consider system (2.2) with initial states* $x(-\tau+1), x(-\tau+2), \cdots, x(0)$, *and the given objective state* $x^* = \delta_{k^n}^\alpha$. *Then system (2.2) is globally stable to* $x^* = \delta_{k^n}^\alpha$, *if and only if*

(i) $[T_{R_0}]_{\alpha,\alpha} = 1$,

(ii) there exists an integer $t^* \in [1 : k^n - 1]$ *such that* $\mathrm{Col}_j(T_{R_{t^*-1}}) \neq \mathbf{0}_{k^n}$, $\forall j \in [1 : k^n]$.

Proof. (Necessity) Suppose that system (2.2) is stable to $x^* = \delta_{k^n}^\alpha$ globally. Then $\delta_{k^n}^\alpha$ is a fixed point of system (2.2), that is, $L\delta_{k^n}^\alpha = \delta_{k^n}^\alpha$, or equivalently, $[L]_{\alpha,\alpha} = 1$. Hence, condition (i) holds.

Since all the initial states $x(-\tau+1), x(-\tau+2), \cdots, x(0) \in \Delta_{k^n}$ can reach $\delta_{k^n}^\alpha$, from the computation of R_t, $t \in \mathbb{N}$, there exists an integer t^* such that $R_{t^*} = \Delta_{k^n}$, which is equivalent to $\mathrm{Col}_j(T_{R_{t^*-1}}) \neq \mathbf{0}_{k^n}$, $\forall j \in [1 : k^n]$.

Let t^* be the smallest positive integer such that $\mathrm{Col}_j(T_{R_{t^*-1}}) \neq \mathbf{0}_{k^n}$, $\forall j \in [1 : k^n]$. Now, we will prove $t^* \leq k^n - 1$. It is enough to show that the number of nonzero columns of $T_{R_{t-1}}$ is $|R_t| \geq t + 1$ for any $t \in [1 : t^*]$.

We use induction to prove it. When $t = 1$, if the number of nonzero columns of T_{R_0} is $|R_1| < 2$, then $(T_{R_0})_{\alpha,\alpha} = 1$ and $(T_{R_0})_{\alpha,j} = 0$, $\forall j \neq \alpha$, and hence $R_{t^*} = \{\delta_{k^n}^\alpha\}$ by Case 2, which is a contradiction.

Now assume that the number of nonzero columns of $T_{R_{t-1}}$ is $|R_t| \geq t + 1$ for some $1 < t \leq t^*$. Since $[T_{R_0}]_{\alpha,\alpha} = 1$, Case 1 shows that $T_{R_{t-1}} \leq T_{R_t}$. Then, $|R_{t+1}| \geq |R_t| \geq t+1$. If $|R_{t+1}| < t+2$, then $|R_{t+1}| = |R_t| = t+1$. Thus, $R_t = R_{t+1}$, which implies $T_{R_{t-1}} = T_{R_t}$ by Case 3. Hence, the number of nonzero columns of $T_{R_{t^*}}$ and $T_{R_{t-1}}$ is equal, which contradicts the minimality of t^*. Therefore, $k^n = |R_{t^*}| \geq t^* + 1$, that is $t^* \leq k^n - 1$.

(Sufficiency) The proof of sufficiency is obvious and we omit this part. □

2.4 Event-triggered control for stabilization of DKVLCNs

Definition 2.4.1. *DKVLCN (2.6) is said to be globally stabilizable to a given state* $x^* = \delta_{k^n}^{\alpha} \in \Delta_{k^n}$*, if for any initial states* $x(-\tau+1), x(-\tau+2), \cdots, x(0) \in \Delta_{k^n}$*, there exist a state feedback event-triggered control* $\overline{u}(t) = Hx(t)$ *and an integer* $T^* \geq 0$*, such that* $x(t) = \delta_{k^n}^{\alpha}$*,* $\forall t \geq T^*$*.*

Remark 2.4.1. *If the given state* $x^* \in \Delta_{k^n}$ *is reachable from some initial state, then there exists a state feedback control, such that it can be reachable in at most* $(k^n - 1)$ *steps.*

In the following, we will study how to design $H \in \mathcal{L}_{(k^m+1)\times k^n}$ for the stabilization problem of DKVLCNs.

First, split the matrix $\overline{L}$ into $(k^m + 1)$ equal blocks:

$$\overline{L} = [\overline{L}_1 \ \overline{L}_2 \ \cdots \ \overline{L}_{k^m+1}], \tag{2.12}$$

where $\overline{L}_i \in \mathcal{L}_{k^n\times k^n}$, $i \in [1 : k^m + 1]$. Then, Algorithm 1 can be utilized to design all the state feedback event-triggered controls via the truth matrices method.

Algorithm 1 Constructing event-triggered state feedback stabilizers

Step 0: Let $R_0 = \{\delta_{k^n}^{\alpha}\}$ and construct the truth matrix $T_{R_0|R_0} \in \mathfrak{B}_{(k^m+1)\times k^n}$:

$$[T_{R_0|R_0}]_{i,j} = \begin{cases} 1, \ \text{if } j = \alpha \text{ and } [\overline{L}_i]_{\alpha,\alpha} = 1, \\ 0, \ \text{otherwise.} \end{cases} \tag{2.13}$$

Check whether $\text{Col}_{\alpha}(T_{R_0|R_0}) \neq \mathbf{0}_{k^m+1}$. If $\text{Col}_{\alpha}(T_{R_0|R_0}) = \mathbf{0}_{k^m+1}$, $\delta_{k^n}^{\alpha}$ is not a (control) fixed point, and the construction problem of stabilizers is not solvable, stop the algorithm. Otherwise, construct the truth matrix $\overline{T}_{R_0|R_0} \in \mathfrak{B}_{(k^m+1)\times k^n}$:

$$\text{Col}_j(\overline{T}_{R_0|R_0}) = \begin{cases} \delta_{k^m+1}^{k^m+1}, & \text{if } (T_{R_0|R_0})_{k^m+1,j} = 1, \\ \text{Col}_j(T_{R_0|R_0}), & \text{otherwise,} \end{cases} \tag{2.14}$$

and go to Step 1.

Step 1: Let $W_1 = \Delta_{k^n} \backslash R_0$ and construct the truth matrix $T_{R_0|W_1} \in \mathfrak{B}_{(k^m+1)\times k^n}$:

$$[T_{R_0|W_1}]_{i,j} = \begin{cases} 1, \ \text{if } (\overline{L}_i)_{\alpha,j} = 1, \forall \delta_{k^n}^{j} \in W_1, \\ 0, \ \text{otherwise.} \end{cases} \tag{2.15}$$

Compute $R_1 = \{\delta_{k^n}^j \mid \mathrm{Col}_j(T_{R_0|W_1}) \neq \mathbf{0}_{k^m+1}\}$. Check whether $R_1 \neq \emptyset$. If $R_1 = \emptyset$, the construction problem of stabilizers is not solvable and stop the algorithm. Otherwise, construct the truth matrix $\overline{T}_{R_0|R_1} \in \mathfrak{B}_{(k^m+1)\times k^n}$:

$$\mathrm{Col}_j(\overline{T}_{R_0|R_1}) = \begin{cases} \delta_{k^m+1}^{k^m+1}, & \text{if } (T_{R_0|W_1})_{k^m+1,j} = 1, \\ \mathrm{Col}_j(T_{R_0|W_1}), & \text{otherwise.} \end{cases} \tag{2.16}$$

If $R_0 \bigcup R_1 = \Delta_{k^n}$, set $t^* = 1$ and go to Step 3; otherwise, go to Step 2.

Step 2: Let $W_t = \Delta_k \backslash [\bigcup_{\lambda=0}^{t-1} R_\lambda]$, where $t \geq 2$. Construct the truth matrix $T_{R_{t-1}|W_t} \in \mathfrak{B}_{(k^m+1)\times k^n}$:

$$[T_{R_{t-1}|W_t}]_{i,j} = \begin{cases} 1, & \text{if } \mathrm{Col}_j(\overline{L}_i) \in R_{t-1}, \forall \delta_{k^n}^j \in W_t, \\ 0, & \text{otherwise.} \end{cases} \tag{2.17}$$

Compute $R_t = \{\delta_{k^n}^j \mid \mathrm{Col}_j(T_{R_{t-1}|W_t}) \neq \mathbf{0}_{k^m+1}\}$. If $R_t = \emptyset$, the construction problem of stabilizers is not solvable and stop the algorithm. Otherwise, construct the truth matrix $\overline{T}_{R_{t-1}|R_t} \in \mathfrak{B}_{(k^m+1)\times k^n}$:

$$\mathrm{Col}_j(\overline{T}_{R_{t-1}|R_t}) = \begin{cases} \delta_{k^m+1}^{k^m+1}, & \text{if } (T_{R_{t-1}|W_t})_{k^m+1,j} = 1, \\ \mathrm{Col}_j(T_{R_{t-1}|W_t}), & \text{otherwise.} \end{cases} \tag{2.18}$$

If

$$\bigcup_{\lambda=0}^{t} R_\lambda = \Delta_{k^n}, \tag{2.19}$$

the stabilization problem is solvable. Denote the minimum number such that (2.19) holds as t^* and go to Step 3; otherwise, set $t = t + 1$. If $t > k^n - 1$, the construction problem of stabilizers is not solvable and stop the algorithm. Otherwise, go to Step 2.

Step 3: The event-triggered state feedback stabilizers $H \in \mathcal{L}_{(k^m+1)\times k^n}$ can be constructed as follows:

$$\begin{cases} H_{|R_0} \leq \overline{T}_{R_0|R_0}, \\ H_{|R_t} \leq \overline{T}_{R_{t-1}|R_t}, \ t \in [1 : t^*]. \end{cases} \tag{2.20}$$

Theorem 2.4.1. *DKVLCN (2.6) is globally stabilizable to $x^* = \delta_{k^n}^\alpha$, if and only if Algorithm 1 reaches Step 3.*

Proof. (Sufficiency) Assume Algorithm 1 reaches Step 3, then we prove that DKVLCN (2.6) is globally stabilizable to $\delta_{k^n}^\alpha$ under the state feedback event-triggered control $\overline{u}(t) = Hx(t)$, where the state feedback gain matrix $H \in \mathcal{L}_{(k^m+1)\times k^n}$ is given by (2.20). From Step 0, when $\delta_{k^n}^{\alpha}$ is a fixed point of system (2.6), there exists a control $H\delta_{k^n}^{\alpha} = \delta_{k^m+1}^{k^m+1}$, such that

$$\begin{aligned}\overline{L}\delta_{k^m+1}^{k^m+1}\delta_{k^n}^{\alpha} &= L\delta_{k^n}^{\alpha}\\ &= \delta_{k^n}^{\alpha}\end{aligned}$$

When $\delta_{k^n}^{\alpha}$ is a control fixed point of system (2.6), there exists at least a control $H\delta_{k^n}^{\alpha} = \delta_{k^m+1}^{h_\alpha}$, where $h_\alpha \in [1 : k^m]$, such that

$$\begin{aligned}\overline{L}\delta_{k^m+1}^{h_\alpha}\delta_{k^n}^{\alpha} &= \widetilde{L}\delta_{k^m}^{h_\alpha}\delta_{k^n}^{\alpha}\\ &= \delta_{k^n}^{\alpha}\end{aligned}$$

From Step 1, for any $x(t_0) = \delta_{k^n}^{j_1} \in R_1$, $x(t_0) \in \{x(-\tau+1), x(-\tau+2), \cdots, x(0)\}$, there exists at least a control $H\delta_{k^n}^{j_1} = \delta_{k^m+1}^{h_1}$, such that

$$\begin{aligned}x(t_0+\tau) &= \overline{L}\overline{u}(t_0)x(t_0)\\ &= \overline{L}\delta_{k^m+1}^{h_1}\delta_{k^n}^{j_1}\\ &= \delta_{k^n}^{\alpha}\end{aligned}$$

If $h_1 = k^m + 1$, then $x(t_0+\tau) = L\delta_{k^n}^{j_1} = \delta_{k^n}^{\alpha}$, that is, $\delta_{k^n}^{j_1}$ can naturally evolve to $\delta_{k^n}^{\alpha}$ in one step. If $h_1 \in [1 : k^m]$, then $x(t_0+\tau) = \widetilde{L}\delta_{k^m}^{h_1}\delta_{k^n}^{j_1} = \delta_{k^n}^{\alpha}$, that is, $\delta_{k^n}^{j_1}$ can be driven to $\delta_{k^n}^{\alpha}$ in one step. No matter in which case, $x(t_0+\rho\tau) = \delta_{k^n}^{\alpha}$, $\forall \rho \in \mathbb{Z}_+$.

Similarly, for any $x(t_0) = \delta_{k^n}^{j_t} \in R_t$, $t \in [2 : t^*]$, $x(t_0) \in \{x(-\tau+1), x(-\tau+2), \cdots, x(0)\}$, there exists at least a control $H\delta_{k^n}^{j_t} = \delta_{k^m+1}^{h_t}$, such that

$$\begin{aligned}x(t_0+\tau) &= \overline{L}\overline{u}(t_0)x(t_0)\\ &= \overline{L}\delta_{k^m+1}^{h_t}\delta_{k^n}^{j_t}\\ &\in R_{t-1}\end{aligned}$$

If $h_t = k^m + 1$, then $x(t_0+\tau) = L\delta_{k^n}^{j_t} \in R_{t-1}$, that is, $\delta_{k^n}^{j_t}$ can naturally evolve to R_{t-1} in one step. If $h_t \in [1 : k^m]$, then $x(t_0+\tau) = \widetilde{L}\delta_{k^m}^{h_t}\delta_{k^n}^{j_t} \in R_{t-1}$, that is, $\delta_{k^n}^{j_t}$ can be driven to R_{t-1} in one step. No matter in which case, there exists a positive integer $\widehat{\rho}$, such that $x(t_0+\rho\tau) = \delta_{k^n}^{\alpha}$, $\forall \rho \geq \widehat{\rho}$ and $\rho \in \mathbb{Z}_+$.

If (2.19) holds when $t = t^*$, then any initial state $x(t_0) \in \{x(-\tau+1), x(-\tau+2), \cdots, x(0)\}$ can reach $\delta_{k^n}^{\alpha}$ in t^* steps. Therefore, DKVLCN (2.6) is globally stabilizable to $x^* = \delta_{k^n}^{\alpha}$.

(Necessity) We prove it by contradiction. Suppose that DKVLCN (2.6) is globally stabilizable to $x^* = \delta_{k^n}^{\alpha}$, but the Algorithm 1 never reach Step 3. That implies that equation (2.19) does not hold until $t = k^n - 1$. Assume $R_{k^n-1} \neq \emptyset$ and

$$\bigcup_{\lambda=0}^{k^n-1} R_\lambda \neq \Delta_{k^n}.$$

Therefore, there exists a state $\overline{x} \in \Delta_{k^n} \backslash [\bigcup_{\lambda=0}^{k^n-1} R_\lambda]$ that cannot reach $[\bigcup_{\lambda=0}^{k^n-1} R_\lambda]$ in $(k^n - 1)$ steps. Hence, state $\overline{x}$ can not reach $\delta_{k^n}^{\alpha}$, which contradicts the condition that delayed system (2.6) is stabilizable to $\delta_{k^n}^{\alpha}$. □

Corollary 2.4.2. *System (2.6) is globally stabilizable to $x^* = \delta_{k^n}^{\alpha}$ under the state feedback event-triggered controller $\overline{u}(t) = Hx(t)$, if and only if there exists an integer $t^* \in [1 : k^n - 1]$, such that*

$$\mathrm{Col}_i(\overline{T}) \neq \mathbf{0}_{k^m+1}, \ \forall i \in [1 : k^n],$$

where $\overline{T} = \overline{T}_{R_0|R_0} + \sum_{\lambda=1}^{t^*} \overline{T}_{\overline{R}_{\lambda-1}|R_\lambda}$. Moreover, all the event-triggered state feedback gain matrices $H \in \mathcal{L}_{(k^m+1)\times k^n}$ under which system (2.6) is globally stabilizable to x^* can be characterized as

$$H \leq \overline{T},$$

and the triggering event set is given by $\Gamma = \Delta_{k^n} \backslash \{\delta_{k^n}^{j} | \mathrm{Col}_j(H) = \delta_{k^m+1}^{k^m+1}\}$.

2.5 An illustrative example

In this section, we provide an illustrative example to demonstrate the applicability of the results obtained in this chapter.

Example 2.5.1. *Consider the following delayed Kleene-Dienes type three-valued logical control networks under event-triggered control:*

$$\begin{cases} X_1(t+1) = X_1(t-1) \wedge_3 X_2(t-1), \\ X_2(t+1) = X_1(t-1) \rightarrow_3 X_2(t-1), \end{cases} \tag{2.21}$$

$$\begin{cases} X_1(t+1) = U(t-1) \rightarrow_3 (X_1(t-1) \wedge_3 X_2(t-1)), \\ X_2(t+1) = U(t-1) \wedge_3 (X_1(t-1) \rightarrow_3 X_2(t-1)). \end{cases} \tag{2.22}$$

When the control input is triggered for certain states, system (2.22) works; otherwise, the evolution follows (2.21).

The algebraic formulations of system (2.21) and (2.22) are given, respectively by:

$$x(t+1) = Lx(t-1), \tag{2.23}$$

$$x(t+1) = \widetilde{L}u(t-1)x(t-1), \tag{2.24}$$

where $x(t) = x_1(t) \ltimes x_2(t) \in \Delta_9$, $u(t) \in \Delta_3$, and

$$L = \delta_9[1\ 5\ 9\ 4\ 5\ 8\ 7\ 7\ 7],$$

$$\widetilde{L} \;=\; \delta_9[1\ 5\ 9\ 4\ 5\ 8\ 7\ 7\ 7\ 2\ 5\ 6\ 5\ 5\ 5\ 5\ 5\ 5\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3].$$

The dynamics under the event-triggered control can be re-expressed as

$$x(t+1) = \overline{L}\overline{u}(t-1)x(t-1), \tag{2.25}$$

where $\overline{L} = [\widetilde{L}\ L] \in \mathcal{L}_{9\times 36}$ and $\overline{u}(t) \in \Delta_4$. Now, we can design all the event-triggered state feedback controls such that (2.25) is globally stabilizable to the state $x^* = \delta_9^7$.

First, according to Algorithm 1, let $R_0 = \{\delta_9^7\}$. The truth matrix $T_{R_0|R_0} \in \mathfrak{B}_{4\times 9}$ is given by

$$T_{R_0|R_0} = \begin{bmatrix} 0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0 \end{bmatrix}. \tag{2.26}$$

From (2.26), $x^* = \delta_9^7$ is a fixed point and the truth matrix $\overline{T}_{R_0|R_0} \in \mathfrak{B}_{4\times 9}$ is given by

$$\overline{T}_{R_0|R_0} = \begin{bmatrix} 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0 \end{bmatrix}. \tag{2.27}$$

Then, denote $W_1 = \Delta_9 \backslash R_0$ and the truth matrix $T_{R_0|W_1} \in \mathfrak{B}_{4\times 9}$ is given by

$$T_{R_0|W_1} = \begin{bmatrix} 0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&1 \end{bmatrix}. \tag{2.28}$$

From (2.28),

$$R_1 = \{\delta_9^8, \delta_9^9\}.$$

Next, construct the truth matrix $\overline{T}_{R_0|R_1} \in \mathfrak{B}_{4\times 9}$

$$\overline{T}_{R_0|R_1} = \begin{bmatrix} 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&1 \end{bmatrix}. \tag{2.29}$$

Then, compute $W_2 = \Delta_9 \backslash (R_0 \bigcup R_1)$ and construct the truth matrix $T_{R_1|W_2} \in \mathfrak{B}_{4\times 9}$

$$T_{R_1|W_2} = \begin{bmatrix} 0&0&1&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&1&0&0&0 \end{bmatrix}. \tag{2.30}$$

From (2.30),

$$R_2 = \{\delta_9^3, \delta_9^6\}.$$

Next, construct the truth matrix $\overline{T}_{R_1|R_2} \in \mathfrak{B}_{4\times 9}$

$$\overline{T}_{R_1|R_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{2.31}$$

Furthermore, compute $W_3 = \Delta_9 \backslash (R_0 \bigcup R_1 \bigcup R_2)$ and construct the truth matrix $T_{R_2|W_3} \in \mathfrak{B}_{4\times 9}$

$$T_{R_2|W_3} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.32}$$

From (2.32),

$$R_3 = \{\delta_9^1, \delta_9^2, \delta_9^4, \delta_9^5\}.$$

Now, construct the truth matrix $\overline{T}_{R_2|R_3} = T_{R_2|W_3}$.

Based on the above discussion, we have

$$\begin{aligned} \overline{T} &= \overline{T}_{R_0|R_0} + \sum_{\lambda=1}^{3} \overline{T}_{R_{\lambda-1}|R_\lambda} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \end{aligned}$$

It is obvious that for $t^* = 3$ we have that all the columns of matrix $\overline{T}$ are nonzero. Therefore, system (2.25) is globally stabilizable to δ_9^7 under the event-triggered control $\overline{u}(t) = Hx(t)$, where the state feedback gain matrix $H \in \mathcal{L}_{4\times 9}$ is given by

$$H = \delta_4[3\ 3\ 4\ 3\ 3\ 4\ 4\ 4\ 4].$$

Besides, the triggering event set $\Gamma = \{\delta_9^1, \delta_9^2, \delta_9^4, \delta_9^5\}$.

2.6 Conclusions

In this chapter, the stability and event-triggered control design of DKVLNs have been investigated. We derived the necessary and sufficient conditions to detect the stability of

KVLNs with time delays. Furthermore, we presented necessary and sufficient conditions for the stabilization of DKVLNs, and designed all the event-triggered state feedback controllers which can stabilize the DKVLNs to a desired state. Finally, an example was given to illustrate the effectiveness of the results obtained in this chapter.

Chapter 3

Robust Control Invariance and Robust Set Stabilization of Mix-Valued Logical Control Networks

3.1 Introduction

Compared with BN and KVLN, MVLN is a more general and representative finite-valued logical network. In an MVLN, each state takes value from a finite set of different size, and in order to manipulate MVLN, control inputs are introduced. In addition, the MVLCNs have been widely used to characterize the control problems of practical networks [7, 143, 46].

It is well known that the performance of practical MVLCNs may be influenced by internal or external ubiquitous disturbances. This is true, for example, in GRNs the disturbances may be the gene mutation, the duplication or deletion of fragments phenomenon in genetic recombination, and external environmental stimuli *etc.* These disturbance inputs often make the system unstable [109], as in case of cancer, which can be regarded as the failure of organism in resisting uncertainties including gene mutation. Therefore, it is meaningful to design a suitable control scheme such that the system with disturbance is robustly stabilized to a singleton state or a state set. Some fundamental and important results on robust stabilization or robust set stabilization problems [67, 127, 71, 81, 56] are available in the scientific literatures. For the robust control invariance, the existing works only concentrate on the criteria to determine whether or not a given set is a RCIS and provide algorithms to design all possible state feedback gain matrices to keep the robust control invariance of the given set. Moreover, for the robust set stabilization, the only existing results concern how to make the system stabilize to a pre-assumed RCIS. Besides, the existing models for the robust control invariance and the robust set stabilization problems are limited to BCNs or KVLNs, and these two problems for MVLCNs are not studied in the necessary detail.

This chapter investigates the robust control invariance and the robust set stabilization problems of MVLCNs. Moreover, a state feedback control is designed if the robust set stabilization problem is solvable. The main contributions of this chapter are:

- An algorithm is proposed to compute the LRCIS of any given set.
- Necessary and sufficient conditions for the robust set stabilization of MVLCNs are derived, and a constructive procedure is presented to design all the time-optimal state feedback controls.
- It is shown that the obtained results can be used to effectively deal with robust partial stabilization problem of MVLCNs.

Consider the following MVLCNs with disturbance inputs:

$$\begin{cases} X_1(t+1) = f_1(X(t); U(t); \Xi(t)), \\ X_2(t+1) = f_2(X(t); U(t); \Xi(t)), \\ \qquad \vdots \\ X_n(t+1) = f_n(X(t); U(t); \Xi(t)), \end{cases} \tag{3.1}$$

where $X(t) = (X_1(t), X_2(t), \cdots, X_n(t))$ are the states, $U(t) = (U_1(t), U_2(t), \cdots, U_m(t))$ are the control inputs, $\Xi(t) = (\Xi_1(t), \Xi_2(t), \cdots, \Xi_q(t))$ are the disturbance inputs; $X_l(t) \in \mathcal{D}_{k_l}$, $l \in [1:n]$, $U_j(t) \in \mathcal{D}_{v_j}$, $j \in [1:m]$ and $\Xi_i(t) \in \mathcal{D}_{w_i}$, $i \in [1:q]$.

Let x_l, u_j, and ξ_i be the vector form of X_l, U_j and Ξ_i respectively. Then in the light of the matrix expression, system (3.1) can be expressed in the following algebraic form:

$$x(t+1) = L\xi(t)u(t)x(t), \tag{3.2}$$

where $L \in \mathcal{L}_{k\times k\nu\omega}$, $x(t) = \ltimes_{l=1}^{n} x_l(t) \in \Delta_k$, $u(t) = \ltimes_{j=1}^{m} u_j(t) \in \Delta_\nu$, $\xi(t) = \ltimes_{i=1}^{q} \xi_i(t) \in \Delta_\omega$, $k = \prod_{l=1}^{n} k_l$, $\nu = \prod_{j=1}^{m} \nu_j$ and $\omega = \prod_{i=1}^{q} \omega_i$.

Now, consider a state feedback control in the form of

$$\begin{cases} U_1(t) = h_1(X_1(t), X_2(t), \cdots, X_n(t)), \\ U_2(t) = h_2(X_1(t), X_2(t), \cdots, X_n(t)), \\ \vdots \\ U_m(t) = h_m(X_1(t), X_2(t), \cdots, X_n(t)), \end{cases} \tag{3.3}$$

whose algebraic form is

$$u(t) = Hx(t), \tag{3.4}$$

where $H \in \mathcal{L}_{\nu \times k}$ is called the state feedback gain matrix, such that system (3.2) is robustly stabilizable to the target set.

Remark 3.1.1. *There is a standard procedure to transfer any finite-valued logical function into its algebraic formulation. One can refer to [14] for more details.*

3.2 Computation of RCIS

In this section, a novel algorithm to compute the LRCIS of a given set is proposed. Moreover, all the possible state feedback controllers under which the obtained set is the LRCIS are determined.

From the definition of RCIS of BCNs [71], the following definitions follow.

Definition 3.2.1. *(RCIS) A nonempty set $\mathcal{S} \subseteq \Delta_k$ is called a RCIS of MVLCNs (3.2), if for any $x(t) \in \mathcal{S}$, there exists at least a control $u(t) \in \Delta_\nu$, such that $x(t+1) \in \mathcal{S}$, $\forall \xi(t) \in \Delta_\omega$.*

Definition 3.2.2. *(LRCIS) The subset $I_c(\mathcal{S})$ is called the LRCIS of $\mathcal{S}$, if it is a RCIS of $\mathcal{S}$, and each RCIS of $\mathcal{S}$ is a subset of $I_c(\mathcal{S})$.*

In the following, for a given nonempty set $\mathcal{S} \subseteq \Delta_k$, we will discuss how to compute its LRCIS.

First, we need to consider the algebraic expression (3.2) of MVLCNs. Let us split $L \in \mathcal{L}_{k \times k\nu\omega}$ into ω equal blocks as

$$L = [L_1 \ L_2 \ \cdots \ L_\omega], \tag{3.5}$$

where $L_i \in \mathcal{L}_{k \times k\nu}$, $i \in [1:\omega]$. Additionally, split each L_i into ν equal blocks

$$L_i = [L_i^1 \ L_i^2 \ \cdots \ L_i^\nu], \tag{3.6}$$

where $L_i^j \in \mathcal{L}_{k\times k}$, $j \in [1:\nu]$.

A truth matrix $T_{\mathcal{S}|\mathcal{S}} \in \mathfrak{B}_{\nu\times k}$ is given by

$$[T_{\mathcal{S}|\mathcal{S}}]_{j,l} = \begin{cases} 1, \text{ if } L_i^j\delta_k^l \in \mathcal{S}, \ \forall i \in [1:\omega], \ \forall \delta_k^l \in \mathcal{S}, \\ 0, \text{ otherwise.} \end{cases} \tag{3.7}$$

Let $\mathcal{S}_1 = \{\delta_k^l \mid \mathrm{Col}_l(T_{\mathcal{S}|\mathcal{S}}) \neq \mathbf{0}_\nu\}$. It is obvious that $\mathcal{S}_1 \subseteq \mathcal{S}$. If $\mathcal{S}_1 = \emptyset$, then $I_c(\mathcal{S}) = \emptyset$. Otherwise, for any $\delta_k^l \in \mathcal{S}_1$, there must exist at least a control δ_ν^j, such that $\mathcal{S}$ is one step robustly reachable from δ_k^l. If $\mathcal{S}_1 = \mathcal{S}$, then $\mathcal{S}$ is a RCIS and $I_c(\mathcal{S}) = \mathcal{S}$. It is obvious that also the reverse holds.

Otherwise, if $\mathcal{S}_1 \subsetneqq \mathcal{S}$, then a truth matrix $T_{\mathcal{S}_1|\mathcal{S}_1} \in \mathfrak{B}_{\nu\times k}$ can be constructed as follows:

$$[T_{\mathcal{S}_1|\mathcal{S}_1}]_{j,l} = \begin{cases} 1, \text{ if } L_i^j\delta_k^l \in \mathcal{S}_1, \ \forall i \in [1:\omega], \ \forall \delta_k^l \in \mathcal{S}_1, \\ 0, \text{ otherwise.} \end{cases} \tag{3.8}$$

Compute now $\mathcal{S}_2 = \{\delta_k^l \mid \mathrm{Col}_l(T_{\mathcal{S}_1|\mathcal{S}_1}) \neq \mathbf{0}_\nu\}$. It is obvious that $\mathcal{S}_2 \subseteq \mathcal{S}_1$. If $\mathcal{S}_2 = \emptyset$, then $I_c(\mathcal{S}) = \emptyset$. Otherwise, for any $\delta_k^l \in \mathcal{S}_2$, there must exist at least a control δ_ν^j, such that $\mathcal{S}_1$ is one step robustly reachable from δ_k^l. If $\mathcal{S}_2 = \mathcal{S}_1$, then for any $x(t) = \delta_k^l \in \mathcal{S}_1$, there exists a control $u(t) = \delta_\nu^j$, such that

$$x(t+1) = L\delta_\omega^i\delta_\nu^j\delta_k^l = L_i^j\delta_k^l \in \mathcal{S}_1, \forall i \in [1:\omega].$$

That implies that $\mathcal{S}_1$ is a RCIS contained in $\mathcal{S}$. Thus, $\mathcal{S}_1 \subseteq I_c(\mathcal{S})$. On the other hand, for any $\widetilde{\delta_k^l} \in (\mathcal{S}\setminus\mathcal{S}_1)$, we can not find a control such that $\mathcal{S}$ is one step robustly reachable from $\widetilde{\delta_k^l}$, *i.e.* $\widetilde{\delta_k^l} \notin I_c(\mathcal{S})$, $\forall\widetilde{\delta_k^l} \in (\mathcal{S}\setminus\mathcal{S}_1)$. That is, $I_c(\mathcal{S})\bigcap(\mathcal{S}\setminus\mathcal{S}_1) = \emptyset$. Furthermore,

$$I_c(\mathcal{S}) \subseteq \mathcal{S}_1 \bigcup (\mathcal{S}\setminus\mathcal{S}_1).$$

Thus, $I_c(\mathcal{S}) \subseteq \mathcal{S}_1$ and therefore, $I_c(\mathcal{S}) = \mathcal{S}_1$.

Otherwise, the above procedure can be executed. Since $\mathcal{S}$ is a set with finite number of states, there must exist an integer $t \in [1:|\mathcal{S}|]$, such that $\mathcal{S}_t = \mathcal{S}_{t-1}$ and $\mathcal{S}_{t'} \neq \emptyset$, $\forall t' \leq t$, where $\mathcal{S}_0 = \mathcal{S}$. Therefore $I_c(\mathcal{S}) = \mathcal{S}_{t-1}$, if $\mathcal{S}_{t'} = \emptyset$, then $I_c(\mathcal{S}) = \emptyset$.

Remark 3.2.1. *From the computation of set $\mathcal{S}_1$, the necessary and sufficient conditions to detect whether or not a given set $\mathcal{S}$ is a RCIS can be immediately derived.*

Based on the above discussion, we propose the following algorithm to compute the LRCIS of $\mathcal{S}$.

Algorithm 2 Computation of the LRCIS

Step 0: Set $\mathcal{S}_0 = \mathcal{S}$.
Step 1: For $t \geq 1$, construct the truth matrix $T_{\mathcal{S}_{t-1}|\mathcal{S}_{t-1}} \in \mathfrak{B}_{\nu\times k}$:

$$[T_{\mathcal{S}_{t-1}|\mathcal{S}_{t-1}}]_{j,l} = \begin{cases} 1, \text{ if } L_i^j \delta_k^l \in \mathcal{S}_{t-1}, \ \forall i \in [1:\omega], \ \forall \delta_k^l \in \mathcal{S}_{t-1}, \\ 0, \text{ otherwise.} \end{cases} \tag{3.9}$$

Compute $\mathcal{S}_t = \{\delta_k^l \mid \mathrm{Col}_l(T_{\mathcal{S}_{t-1}|\mathcal{S}_{t-1}}) \neq \mathbf{0}_\nu\}$. If $\mathcal{S}_t = \emptyset$, then $I_c(\mathcal{S}) = \emptyset$ and stop the algorithm. Otherwise, check whether

$$\mathcal{S}_t = \mathcal{S}_{t-1}. \tag{3.10}$$

If (3.10) holds, denote the minimum number such that (3.10) holds as $\widehat{t}$, and go to Step 2. Otherwise, let $t = t+1$. If $t > |\mathcal{S}|$, then $I_c(\mathcal{S}) = \emptyset$, stop the algorithm, otherwise, go to Step 1.
Step 2: The LRCIS contained in $\mathcal{S}$ is obtained as follows

$$I_c(\mathcal{S}) = \mathcal{S}_{\widehat{t-1}}. \tag{3.11}$$

Remark 3.2.2. *Compared with the robust control invariance proposed in [71], Algorithm 2 does not only provide the method to compute the LRCIS of MVLCNs (3.2), but also determine all the possible state feedback controllers.*

The proof of the following corollary is obvious.

Corollary 3.2.1. *The nonempty LRCIS of system (3.2) is $\mathcal{S}_{\widehat{t-1}}$, if and only if the nonzero columns of truth matrices $T_{\mathcal{S}_{\widehat{t-1}}|\mathcal{S}_{\widehat{t-1}}}$ and $T_{\mathcal{S}_{\widehat{t-2}}|\mathcal{S}_{\widehat{t-2}}}$ are identical. Moreover, all possible state feedback gain matrices $H \in \mathcal{L}_{\nu\times k}$ under which $\mathcal{S}_{\widehat{t-1}}$ is the LRCIS can be characterized as*

$$\begin{cases} H|_{\mathcal{S}_{\widehat{t-1}}} \leq T_{\mathcal{S}_{\widehat{t-1}}|\mathcal{S}_{\widehat{t-1}}}, \\ H|_{(\Delta_k \setminus \mathcal{S}_{\widehat{t-1}})} \leq \mathbf{1}_{\nu\times k}|_{(\Delta_k \setminus \mathcal{S}_{\widehat{t-1}})}. \end{cases} \tag{3.12}$$

3.3 Robust set stabilization of MVLCNs

In this section, necessary and sufficient conditions for the robust set stabilization of MVLCNs based on RCIS are established, and an algorithm to design all the time-optimal state feedback controls via antecedence solution technique is presented.

From the definition of robust set stabilization of BCNs [81], the following definition follows.

Definition 3.3.1. *(Robust set stabilization) The MVLCN (3.2) is said to be robustly stabilizable to the nonempty set $\mathcal{S} \subseteq \Delta_k$, if for any initial state $x(0) \in \Delta_k$, there exist a state feedback control $u(t) = Hx(t)$ and an integer $\tau \geq 0$, such that $x(t) \in \mathcal{S}$, $\forall t \geq \tau$ and $\{\xi(t) : t \in \mathbb{N}\} \subseteq \Delta_\omega$.*

Lemma 3.3.1. *The MVLCN (3.2) is robustly stabilizable to $\mathcal{S}$, if and only if it is robustly stabilizable to $I_c(\mathcal{S})$.*

The proof of Lemma 3.3.1 is trivial.

Remark 3.3.1. *If the set $I_c(\mathcal{S})$ is robustly reachable from some initial state, then there exists a state feedback control, such that it can be robustly reachable in at most $(k - |I_c(\mathcal{S})|)$ steps.*

In the following, we will study how to design $H \in \mathcal{L}_{\nu \times k}$ for the robust set stabilization problem of MVLCNs based on the algebraic forms (3.2) and (3.4).

First, for a given set $\mathcal{S} \subseteq \Delta_k$, we compute its LRCIS applying Algorithm 2. When $I_c(\mathcal{S}) \neq \emptyset$, then Algorithm 3 can be used to design all the time-optimal state feedback controls via antecedence solution technique.

Algorithm 3 Constructing time-optimal state feedback stabilizers

---

Step 1: Let $\overline{W}_0 = I_c(\mathcal{S})$, $W_1 = \Delta_k \backslash \overline{W}_0$ and construct the truth matrix $T_{\overline{W}_0|W_1} \in \mathfrak{B}_{\nu \times k}$:

$$[T_{\overline{W}_0|W_1}]_{j,l} = \begin{cases} 1, \text{ if } L_i^j \delta_k^l \in \overline{W}_0, \ \forall i \in [1:\omega], \forall \delta_k^l \in W_1, \\ 0, \text{ otherwise.} \end{cases} \tag{3.13}$$

Compute $R_1(I_c(\mathcal{S})) = \{\delta_k^l \mid \mathrm{Col}_l(T_{\overline{W}_0|W_1}) \neq \mathbf{0}_\nu\}$. Check whether $R_1(I_c(\mathcal{S})) \neq \emptyset$. If $R_1(I_c(\mathcal{S})) = \emptyset$, the construction problem of robust set stabilizers is not solvable and we can stop the algorithm. If $I_c(\mathcal{S}) \bigcup R_1(I_c(\mathcal{S})) = \Delta_k$, set $t^* = 1$ and go to Step 3; otherwise, go to Step 2.

Step 2: Compute $\overline{W}_{t-1} = \bigcup_{\lambda=0}^{t-1} R_\lambda(I_c(\mathcal{S}))$ and $W_t = \Delta_k \backslash \overline{W}_{t-1}$, where $R_0(I_c(\mathcal{S})) = I_c(\mathcal{S})$, $t \geq 2$. Construct the truth matrix $T_{\overline{W}_{t-1}|W_t} \in \mathfrak{B}_{\nu \times k}$:

$$[T_{\overline{W}_{t-1}|W_t}]_{j,l} = \begin{cases} 1, \text{ if } L_i^j \delta_k^l \in \overline{W}_{t-1}, \ \forall i \in [1:\omega], \ \forall \delta_k^l \in W_t, \\ 0, \text{ otherwise.} \end{cases} \tag{3.14}$$

Compute $R_t(I_c(\mathcal{S})) = \{\delta_k^l \mid \mathrm{Col}_l(T_{\overline{W}_{t-1}|W_t}) \neq \mathbf{0}_\nu\}$. If $R_t(I_c(\mathcal{S})) = \emptyset$, the construction problem of robust set stabilizers is not solvable and we can stop the algorithm. If

$$\bigcup_{\lambda=0}^{t} R_\lambda(I_c(\mathcal{S})) = \Delta_k, \tag{3.15}$$

the robust set stabilization problem is solvable. Denote the minimum number such that (3.15) holds as t^* and go to Step 3; otherwise, set $t = t + 1$. If $t > k - |I_c(\mathcal{S})|$, the construction problem of robust set stabilizers is not solvable and we can stop the algorithm. Otherwise, go to Step 2.

Step 3: The time-optimal state feedback stabilizers $H \in \mathcal{L}_{\nu \times k}$ can be constructed as follows:

$$\begin{cases} H_{|_{I_c(\mathcal{S})}} \leq T_{I_c(\mathcal{S})|I_c(\mathcal{S})}, \\ H_{|_{R_t(I_c(\mathcal{S}))}} \leq T_{\overline{W}_{t-1}|R_t(I_c(\mathcal{S}))}, \ t \in [1 : t^*], \end{cases} \tag{3.16}$$

where the truth matrix $T_{I_c(\mathcal{S})|I_c(\mathcal{S})}$ is obtained from Algorithm 2 and

$$\mathrm{Col}_l(T_{\overline{W}_{t-1}|R_t(I_c(\mathcal{S}))}) = \begin{cases} \mathrm{Col}_l(T_{\overline{W}_{t-1}|W_t}), \ \text{if } \delta_k^l \in R_t(I_c(\mathcal{S})), \\ \mathbf{0}_{k^m}, \ \text{otherwise}. \end{cases}$$

Algorithm 3 is depicted in Figure 3.1, where $R_\lambda(I_c(\mathcal{S})) \neq \emptyset$, $\lambda \in [1 : t^*]$, and

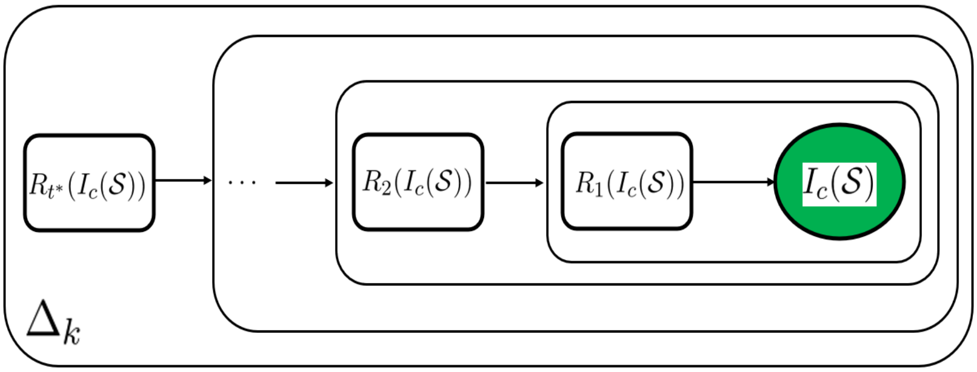

Figure 3.1: Representation of Algorithm 3

$$R_i(I_c(\mathcal{S})) \bigcap R_j(I_c(\mathcal{S})) = \emptyset, \forall i, j \in [0 : t^*], i \neq j,$$

which implies that t^* is the shortest time for all initial states to reach $I_c(\mathcal{S})$ under the disturbances.

Theorem 3.3.2. *The MVLCN (3.2) can be robustly stabilized to $I_c(\mathcal{S})$ by a state feedback control $u(t) = Hx(t)$, if and only if*

$$\bigcup_{\lambda=0}^{t^*} R_\lambda(I_c(\mathcal{S})) = \Delta_k, \tag{3.17}$$

where t^* and $R_\lambda(I_c(\mathcal{S}))$, $\lambda \in [0 : t^*]$ are obtained from Algorithm 3.

Proof. (Sufficiency) Assume condition (3.17) holds, then for any $\delta_k^{l_1} \in R_1(I_c(\mathcal{S}))$, there exists at least a control $H\delta_k^{l_1} = \delta_\nu^{j_1}$ such that $R_0(I_c(\mathcal{S}))$ is one step robustly reachable from $\delta_k^{l_1} \in R_1(I_c(\mathcal{S}))$. Then, for any $\delta_k^{l_2} \in R_2(I_c(\mathcal{S}))$, there exists at least a control $H\delta_k^{l_2} = \delta_\nu^{j_2}$ such that $R_0(I_c(\mathcal{S}))\bigcup R_1(I_c(\mathcal{S}))$ is one step robustly reachable from this state. It implies that state $\delta_k^{l_2}$ can reach $R_1(I_c(\mathcal{S}))$ in one step under arbitrary disturbance, or $\delta_k^{l_2}$ can reach $R_1(I_c(\mathcal{S}))$ in one step under any disturbance in set Γ_1 and $\delta_k^{l_2}$ can reach $R_0(I_c(\mathcal{S}))$ in one step under all the disturbances in Γ_2, where $\Gamma_1\bigcup\Gamma_2 = \Delta_\omega$ is a partition of disturbance inputs. No matter in which case, $I_c(\mathcal{S})$ is robustly reachable from $R_2(I_c(\mathcal{S}))$ after two steps.

Similarly, for any $\delta_k^{l_t} \in R_t(I_c(\mathcal{S}))$, $t \in [3 : t^*]$, there exists at least a control $H\delta_k^{l_t} = \delta_\nu^{j_t}$ such that $\bigcup_{\lambda=0}^{t-1} R_\lambda(I_c(\mathcal{S}))$ is one step robustly reachable from $\delta_k^{l_t} \in R_t(I_c(\mathcal{S}))$. Since there exists a positive integer t^* such that $\bigcup_{\lambda=0}^{t^*} R_\lambda(I_c(\mathcal{S})) = \Delta_k$ holds, then all the states can be robustly steered to $I_c(\mathcal{S})$, and therefore, the system (3.2) is robustly stabilizable to $I_c(\mathcal{S})$.

(Necessity) Suppose now that the MVLCN (3.2) is robustly stabilizable to $I_c(\mathcal{S})$, and assume by contradiction that the equation $\bigcup_{\lambda=0}^{t} R_\lambda(I_c(\mathcal{S})) = \Delta_k$ does not hold until $t = k - |I_c(\mathcal{S})|$. Assume $R_{k-|I_c(\mathcal{S})|}(I_c(\mathcal{S})) \neq \emptyset$ and

$$\bigcup_{\lambda=0}^{k-|I_c(\mathcal{S})|} R_\lambda(I_c(\mathcal{S})) \neq \Delta_k.$$

Then, there exists a state $\widehat{x} \in \Delta_k \backslash [\bigcup_{\lambda=0}^{k-|I_c(\mathcal{S})|} R_\lambda(I_c(\mathcal{S}))]$, such that no control can drive it to $[\bigcup_{\lambda=0}^{k-|I_c(\mathcal{S})|} R_\lambda(I_c(\mathcal{S}))]$ under the influence of disturbances. Hence, state $\widehat{x}$ can not reach $I_c(\mathcal{S})$ under any disturbance, which contradicts the condition that the system (3.2) is robustly stabilizable to $I_c(\mathcal{S})$. □

Based on the above discussion, the following corollary is obvious.

Corollary 3.3.3. *System (3.2) is robustly stabilizable to $\mathcal{S} \subseteq \Delta_k$ under the state feedback controller $u(t) = Hx(t)$, if and only if*

(i) $I_c(\mathcal{S}) \neq \emptyset$,

(ii) there exists an integer $t^* \in [1 : k - |I_c(\mathcal{S})|]$, such that

$$\mathrm{Col}_i(\widehat{T}) \neq \mathbf{0}_\nu, \forall i \in [1 : k],$$

where $\widehat{T} = T_{I_c(\mathcal{S})|I_c(\mathcal{S})} + \sum_{\lambda=1}^{t^*} T_{\overline{W}_{\lambda-1}|R_\lambda(I_c(\mathcal{S}))}$.

Moreover, if (i) and (ii) hold, then for all the time-optimal state feedback gain matrices $H \in \mathcal{L}_{\nu\times k}$ under which the system (3.2) is robustly stabilizable to $\mathcal{S}$ can be characterized as $H \leq \widehat{T}$.

Remark 3.3.2. *The results obtained in this chapter can be utilized to investigate the computation of RCIS, robust set stabilization problem of BCNs or KVLCNs. Moreover, these results can be used to study robust partial stabilization problem of generic logical system.*

3.4 An illustrative example

In this section, we provide an illustrative example to show how to use the proposed method to investigate the robust partial stabilization problem of MVLCNs.

Example 3.4.1. *Consider the following MVLCN:*

$$\begin{cases} X_1(t+1) = [((X_1(t) \wedge_2 X_2(t)) \vee_2 \Xi(t)) \rightarrow_2 U(t)] \\ \qquad\qquad \leftrightarrow_2 [X_1(t) \wedge_2 \neg_2(X_2(t) \leftrightarrow_3 X_3(t))], \\ X_2(t+1) = [(X_1(t) \leftrightarrow_2 (X_2(t) \leftrightarrow_3 X_3(t))) \vee_2 \Xi(t)] \rightarrow_2 U(t), \\ X_3(t+1) = \{[(X_1(t) \leftrightarrow_2 (X_2(t) \leftrightarrow_3 X_3(t))) \vee_2 \Xi(t)] \rightarrow_2 U(t)\} \\ \qquad\qquad \leftrightarrow_3 \{[(X_2(t) \rightarrow_3 \Xi(t)) \vee_3 U(t)] \rightarrow_3 [X_1(t) \leftrightarrow_2 (X_2(t) \leftrightarrow_3 X_3(t))]\}, \end{cases} \tag{3.18}$$

where $X_1, X_2 \in \mathcal{D}$, $X_3 \in \mathcal{D}_3$ are the states, $U \in \mathcal{D}_3$ is the control input, and $\Xi \in \mathcal{D}$ is the disturbance input. For more details about the definitions and notations of logical operators for mix-valued logical variables (refer to [14]).

In the following, we aim to design the time-optimal state feedback controllers such that the states of the first two nodes of system (3.18) are globally convergent to x_e^2 under any disturbance $\xi \in \Delta$, where $x_e^2 = x_1^e \ltimes x_2^e = \delta_4^3$.

Let $\mathcal{M} = \{\delta_4^3 \ltimes \delta_3^l | \ l \in [1:3]\} = \{\delta_{12}^7, \delta_{12}^8, \delta_{12}^9\}$. In this case, the robust partial stabilization problem of MVLCNs (3.18) can be transformed into the robust set stabilization problem.

Using the matrix expression of mix-valued logical function, system (3.18) can be converted to

$$x(t+1) = L\xi(t)u(t)x(t), \tag{3.19}$$

where $x(t) = \ltimes_{l=1}^3 x_l(t) \in \Delta_{12}$, $u(t) \in \Delta_3$, $\xi(t) \in \Delta_2$, and

$$\begin{aligned} L = \delta_{12}[&7\ 8\ 3\ 3\ 2\ 7\ 9\ 8\ 7\ 7\ 8\ 9\ 7\ 8\ 3\ 3\ 2\ 7\ 9\ 8\ 7\ 7\ 8\ 9 \\ &4\ 5\ 12\ 12\ 11\ 4\ 6\ 5\ 4\ 4\ 5\ 6\ 7\ 8\ 3\ 3\ 2\ 7\ 9\ 8\ 7\ 7\ 8\ 9 \\ &7\ 8\ 3\ 3\ 2\ 7\ 9\ 8\ 7\ 7\ 8\ 9\ 4\ 5\ 9\ 3\ 2\ 10\ 9\ 8\ 10\ 10\ 11\ 9]. \end{aligned}$$

First, split L into 2 blocks as $L = [L_1\ L_2]$ and split each L_i, $i \in [1:2]$ into 3 equal blocks as

$$\begin{aligned} L_1^1 &= \delta_{12}[7\ 8\ 3\ 3\ 2\ 7\ 9\ 8\ 7\ 7\ 8\ 9], & L_2^1 &= \delta_{12}[7\ 8\ 3\ 3\ 2\ 7\ 9\ 8\ 7\ 7\ 8\ 9], \\ L_1^2 &= \delta_{12}[7\ 8\ 3\ 3\ 2\ 7\ 9\ 8\ 7\ 7\ 8\ 9], & L_2^2 &= \delta_{12}[7\ 8\ 3\ 3\ 2\ 7\ 9\ 8\ 7\ 7\ 8\ 9], \\ L_1^3 &= \delta_{12}[4\ 5\ 12\ 12\ 11\ 4\ 6\ 5\ 4\ 4\ 5\ 6], & L_2^3 &= \delta_{12}[4\ 5\ 9\ 3\ 2\ 10\ 9\ 8\ 10\ 10\ 11\ 9]. \end{aligned}$$

Then, according to Algorithm 2, let $\mathcal{M}_0 = \mathcal{M}$. The truth matrix $T_{\mathcal{M}_0|\mathcal{M}_0} \in \mathfrak{B}_{3\times 12}$ is given by

$$T_{\mathcal{M}_0|\mathcal{M}_0} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.20}$$

From (3.20),

$$\mathcal{M}_1 = \{\delta_{12}^7, \delta_{12}^8, \delta_{12}^9\} = \mathcal{M}_0.$$

It is obvious that $\mathcal{M}$ is a RCIS. Then, according to Algorithm 3, let $\overline{W}_0 = \mathcal{M}$, $W_1 = \Delta_{12}\backslash\overline{W}_0$ and construct the truth matrix $T_{\overline{W}_0|W_1} \in \mathfrak{B}_{3\times 12}$

$$T_{\overline{W}_0|W_1} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.21}$$

From (3.21),

$$R_1(\mathcal{M}) = \{\delta_{12}^1, \delta_{12}^2, \delta_{12}^6, \delta_{12}^{10}, \delta_{12}^{11}, \delta_{12}^{12}\}.$$

Then, compute $\overline{W}_1 = R_0(\mathcal{M}) \bigcup R_1(\mathcal{M})$ and $W_2 = \Delta_{12}\backslash\overline{W}_1$, where $R_0(\mathcal{M}) = \mathcal{M}$. Construct the truth matrix $T_{\overline{W}_1|W_2} \in \mathfrak{B}_{3\times 12}$

$$T_{\overline{W}_1|W_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.22}$$

From (3.22),

$$R_2(\mathcal{M}) = \{\delta_{12}^3, \delta_{12}^5\}.$$

Furthermore, compute $\overline{W}_2 = R_0(\mathcal{M}) \bigcup R_1(\mathcal{M}) \bigcup R_2(\mathcal{M})$, $W_3 = \Delta_{12}\backslash\overline{W}_2$ and construct the truth matrix $T_{\overline{W}_2|W_3} \in \mathfrak{B}_{3\times 12}$

$$T_{\overline{W}_2|W_3} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.23}$$

From (3.23),

$$R_3(\mathcal{M}) = \{\delta_{12}^4\}.$$

It is obvious that $\bigcup_{\lambda=0}^{3} R_\lambda(\mathcal{M}) = \Delta_{12}$. Then the robust stabilizers of set $\mathcal{M}$ is given by

$$\begin{cases} H_{|\mathcal{M}_0} \leq T_{\mathcal{M}_0|\mathcal{M}_0}, \\ H_{|R_1(\mathcal{M})} \leq T_{\overline{W}_0|R_1(\mathcal{M})}, \\ H_{|R_2(\mathcal{M})} \leq T_{\overline{W}_1|R_2(\mathcal{M})}, \\ H_{|R_3(\mathcal{M})} \leq T_{\overline{W}_2|R_3(\mathcal{M})}, \end{cases} \tag{3.24}$$

where $T_{\overline{W}_0|R_1(\mathcal{M})} = T_{\overline{W}_0|W_1}$, $T_{\overline{W}_1|R_2(\mathcal{M})} = T_{\overline{W}_1|W_2}$ and $T_{\overline{W}_2|R_3(\mathcal{M})} = T_{\overline{W}_2|W_3}$.

From (3.24), it follows that the possible choices of matrix $H \in \mathcal{L}_{3\times 12}$ are

$$\begin{aligned} &\mathrm{Col}_l(H) \in \{\delta_3^1, \delta_3^2\},\ l \in [1:12] \setminus \{3,4,5\}, \\ &\mathrm{Col}_3(H) = \delta_3^3, \\ &\mathrm{Col}_5(H) \in \{\delta_3^1, \delta_3^2, \delta_3^3\}, \\ &\mathrm{Col}_4(H) \in \{\delta_3^1, \delta_3^2, \delta_3^3\}. \end{aligned} \tag{3.25}$$

Based on the above discussion, the states of the first two nodes of the system (3.18) keep $x_e^2 = \delta_4^3$ forever regardless the disturbance inputs under the state feedback control $u(t) = Hx(t)$, where a state feedback gain matrix is given by

$$H = \delta_3[2\ 2\ 3\ 2\ 3\ 1\ 2\ 2\ 1\ 1\ 2\ 2].$$

3.5 Conclusions

In this chapter, we have investigated the robust control invariance and robust set stabilization problems of MVLCNs with disturbance inputs. An algorithm has been proposed to determine the LRCIS for MVLCNs of any given set. Moreover, necessary and sufficient conditions to detect the solvability of robust set stabilization problem of MVLCNs have been derived. Using an antecedence solution technique, a constructive algorithm has been established to design all the time-optimal stabilizers. Finally, an illustrative example has been presented to show the applications of the results obtained in this chapter to robust partial stabilization problem of MVLCNs.

Chapter 4

Robust Set Stability and Set Stabilization of Probabilistic Boolean Control Networks

4.1 Introduction

Recently, set stability and set stabilization have been one of the hot topics in deterministic model [36, 137, 59] and probabilistic model of GRNs using STP method [37, 147, 61]. Set stability of BN implies that all the initial states can reach the only invariant set while set stabilization of BCN means that all the initial states can be steered to the desired control invariant set by control inputs. Several kinds of concepts of set stability and set stabilization of PBCN were provided in [37, 147, 75, 142, 99]. It is obvious that if the target set becomes a single-point set, then the corresponding problems are converted to the usual stability and stabilization problems [74, 75, 73], respectively.

In a real GRN external disturbances are ubiquitous and may lead the network dynamics to some unexpected behaviours [109]. Therefore, it is important to study set stability and set stabilization of PBCN with disturbances. There are some works concerning robust control invariance and robust set stabilization of BCN [67, 127, 71, 81]. Nevertheless, due to the effect of disturbance inputs and stochastic nature of PBCN, the results in BCN are not easily generalized to PBCN. In [83], the robust control invariance of PBCN was investigated via event-triggered control. To the best of author' knowledge, there are no results available on robust set stability and robust set stabilization of PBCN at present.

This chapter investigates the robust set stability of PBN and the robust set stabilization of PBCN. Using the STP method, PBNs and PBCNs with disturbances can be converted into the disturbed stochastic discrete time systems with algebraic forms, based on which the classical control theory and methods can be used to analyze and control logical systems. The novelties are the following:

- The LRIS and LRCIS with probability 1 are calculated for the first time.
- The criteria to determine the finite-time robust set stability and robust set stabilization with probability 1 are firstly derived. The results obtained can be utilized to study several unsolved disturbed PBCN problems, including finite-time robust output tracking, robust synchronization and robust partial stabilization with probability 1.
- A design procedure is proposed to calculate all the time-optimal robust feedback stabilizers via antecedence solution technique. Compared with the traditional design method, the controls can be obtained directly from the nonzero columns of the truth matrices, and the computation involved can be easily executed by Matlab.

A disturbed PBN is a randomly switched Boolean network with disturbances

$$X(t+1) = f^{\sigma(t)}(X(t); \Xi(t)), \tag{4.1}$$

where $X(t) = (X_1(t), X_2(t), \cdots, X_n(t)) \in \mathcal{D}^n$ and $\Xi(t) = (\Xi_1(t), \Xi_2(t), \cdots, \Xi_q(t)) \in \mathcal{D}^q$ are the states and disturbance inputs, respectively. Moreover, $\sigma(t) \in [1:N]$ is a stochastic switching signal and N denotes the number of possible sub-systems. Finally, $f^\nu : \mathcal{D}^{n+q} \longmapsto \mathcal{D}^n$, $\nu \in [1:N]$ is an n-dimensional logical function.

Based on the matrix expression, we identify $1 \sim \delta_2^1$, $0 \sim \delta_2^2$ and $\nu \sim \delta_N^\nu$, where $\nu \in [1:N]$. Then, in the vector form, the disturbed PBN (4.1) becomes

$$x(t+1) = \mathbf{L}\sigma(t)\xi(t)x(t) = [\mathbf{L}_1 \ \mathbf{L}_2 \ \cdots \ \mathbf{L}_N]\sigma(t)\xi(t)x(t), \tag{4.2}$$

where $\mathbf{L}_\nu \in \mathcal{L}_{2^n \times 2^{n+q}}$ is the structural matrix of f^ν, $\nu \in [1:N]$, $x(t) = \ltimes_{i=1}^n x_i(t) \in \Delta_{2^n}$, $\xi(t) = \ltimes_{j=1}^q \xi_j(t) \in \Delta_{2^q}$ and $\sigma(t) \in \Delta_N$. Here, we assume that $\sigma(t)$ is an independent identically distributed process with probability distribution

$$\mathbb{P}\{\sigma(t) = \delta_N^j\} = \mathbb{P}\{\text{subnetwork } j \text{ is selected}\} = \mathbf{p}_j^\sigma, \tag{4.3}$$

where $0 \le \mathbf{p}_j^\sigma \le 1$, $j \in [1:N]$ and $\sum_{j=1}^N \mathbf{p}_j^\sigma = 1$. Denote the probability distribution vector of $\sigma(t)$ as $\mathbf{P}^\sigma = [\mathbf{p}_1^\sigma \ \mathbf{p}_2^\sigma \cdots \mathbf{p}_N^\sigma]^T$. Then the transition probability matrix of disturbed PBN (4.1) is expressed as follows:

$$\mathbf{P} = \mathbf{L} \ltimes \mathbf{P}^\sigma := [\mathbf{P}_1 \ \mathbf{P}_2 \ \cdots \ \mathbf{P}_{2^q}], \tag{4.4}$$

where $[\mathbf{P}_k]_{i,j} = \mathbb{P}\{x(t+1) = \delta_{2^n}^i | x(t) = \delta_{2^n}^j, \xi(t) = \delta_{2^q}^k\}$, $k \in [1:2^q]$ and $0 \leq [\mathbf{P}_k]_{i,j} \leq 1$, $\sum_{i=1}^{2^n} [\mathbf{P}_k]_{i,j} = 1$, $i,j \in [1:2^n]$.

The dynamics of a disturbed PBCN with stochastic switching signals can be denoted by

$$X(t+1) = f^{\sigma(t)}(X(t); \Xi(t); U(t)), \tag{4.5}$$

where $U(t) = (U_1(t), U_2(t), \cdots, U_m(t)) \in \mathcal{D}^m$ are control inputs. Similarly, the disturbed PBCN (4.5) can be converted into the following algebraic formulation

$$x(t+1) = \overline{\mathbf{L}}\sigma(t)\xi(t)u(t)x(t), \tag{4.6}$$

where $\overline{\mathbf{L}} \in \mathcal{L}_{2^n \times N2^{n+m+q}}$ and $u(t) = \ltimes_{l=1}^m u_l(t) \in \Delta_{2^m}$. Furthermore, the control-dependent transition probability matrix of disturbed PBCN (4.6) is expressed as $\overline{\mathbf{P}} = \overline{\mathbf{L}} \ltimes \mathbf{P}^\sigma$. Split now the matrix $\overline{\mathbf{P}}$ into 2^{q+m} equal blocks:

$$\overline{\mathbf{P}} = [\overline{\mathbf{P}}_1^1\ \overline{\mathbf{P}}_1^2\ \cdots\ \overline{\mathbf{P}}_1^{2^m}\ \overline{\mathbf{P}}_2^1\ \overline{\mathbf{P}}_2^2\ \cdots\ \overline{\mathbf{P}}_2^{2^m}\ \cdots\ \overline{\mathbf{P}}_{2^q}^1\ \overline{\mathbf{P}}_{2^q}^2\ \cdots\ \overline{\mathbf{P}}_{2^q}^{2^m}], \tag{4.7}$$

where $[\overline{\mathbf{P}}_k^l]_{i,j} = \mathbb{P}\{x(t+1) = \delta_{2^n}^i | x(t) = \delta_{2^n}^j, \xi(t) = \delta_{2^q}^k, u(t) = \delta_{2^m}^l\}$, $k \in [1:2^q]$, $l \in [1:2^m]$ and $0 \leq [\overline{\mathbf{P}}_k^l]_{i,j} \leq 1$, $\sum_{i=1}^{2^n} [\overline{\mathbf{P}}_k^l]_{i,j} = 1$, $i,j \in [1:2^n]$.

For PBCN (4.6), we consider the state feedback controllers in the form

$$u(t) = Hx(t), \tag{4.8}$$

where $H \in \mathcal{L}_{2^m \times 2^n}$ is called the state feedback gain matrix to achieve our control objectives.

4.2 RIS with probability 1 of PBNs

Definition 4.2.1. *(RIS) A nonempty set $\mathcal{S} \subseteq \Delta_{2^n}$ is called a RIS with probability 1 of PBN (4.2), if for any $x(t) \in \mathcal{S}$ and $\xi(t) \in \Delta_{2^q}$, it follows that $\mathbb{P}\{x(t+1) \in \mathcal{S} | x(t) \in \mathcal{S}\} = 1$.*

Definition 4.2.2. *(LRIS) The subset $I(\mathcal{S})$ is called the LRIS with probability 1 of PBN (4.2) contained in $\mathcal{S}$, if it is a RIS with probability 1, and each RIS with probability 1 in $\mathcal{S}$ is a subset of $I(\mathcal{S})$.*

Definition 4.2.3. *(RCIS) A nonempty set $\mathcal{S} \subseteq \Delta_{2^n}$ is called a RCIS with probability 1 of PBCN (4.6), if for any $x(t) \in \mathcal{S}$ and $\xi(t) \in \Delta_{2^q}$, there exists at least a control $u(t) \in \Delta_{2^m}$, such that $\mathbb{P}\{x(t+1) \in \mathcal{S} | x(t) \in \mathcal{S}\} = 1$.*

Definition 4.2.4. *(LRCIS) The subset $I_c(\mathcal{S})$ is called the LRCIS with probability 1 of PBCN (4.6) contained in $\mathcal{S}$, if it is a RCIS with probability 1, and each RCIS with probability 1 in $\mathcal{S}$ is a subset of $I_c(\mathcal{S})$.*

In the following, for a given nonempty set $\mathcal{S} \subseteq \Delta_{2^n}$, we discuss how to compute its LRIS with probability 1. Here, we only investigate the RIS and RCIS with probability 1, for the convenience of discussion, the statement "with probability 1" is omitted without confusion.

First, a truth matrix $T_{\mathcal{S}|\mathcal{S}} \in \mathfrak{B}_{2^q \times 2^n}$ is constructed, where

$$[T_{\mathcal{S}|\mathcal{S}}]_{k,j} = \begin{cases} 1, \text{ if } \sum_{\delta_{2^n}^i \in \mathcal{S}} [\mathbf{P}_k]_{i,j} = 1, \ \forall \delta_{2^n}^j \in \mathcal{S}, \\ 0, \text{ otherwise.} \end{cases} \tag{4.9}$$

Compute $\mathcal{S}_1 = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T_{\mathcal{S}|\mathcal{S}}) = \mathbf{1}_{2^q}\}$. It is obvious that $\mathcal{S}_1 \subseteq \mathcal{S}$. If $\mathcal{S}_1 = \emptyset$, then $I(\mathcal{S}) = \emptyset$. Otherwise, for any $\delta_{2^n}^j \in \mathcal{S}_1$, $\mathcal{S}$ is one step robustly reachable from $\delta_{2^n}^j$ with probability 1. If $\mathcal{S}_1 = \mathcal{S}$, then $\mathcal{S}$ is a RIS and $I(\mathcal{S}) = \mathcal{S}$. Conversely, if $I(\mathcal{S}) = \mathcal{S}$, then $\mathcal{S}_1 = \mathcal{S}$.

Otherwise, if $\mathcal{S}_1 \subsetneqq \mathcal{S}$, then a truth matrix $T_{\mathcal{S}_1|\mathcal{S}_1} \in \mathfrak{B}_{2^q \times 2^n}$ can be constructed as follows:

$$[T_{\mathcal{S}_1|\mathcal{S}_1}]_{k,j} = \begin{cases} 1, \text{ if } \sum_{\delta_{2^n}^i \in \mathcal{S}_1} [\mathbf{P}_k]_{i,j} = 1, \ \forall \delta_{2^n}^j \in \mathcal{S}_1, \\ 0, \text{ otherwise.} \end{cases} \tag{4.10}$$

Compute now $\mathcal{S}_2 = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T_{\mathcal{S}_1|\mathcal{S}_1}) = \mathbf{1}_{2^q}\}$. It is obvious that $\mathcal{S}_2 \subseteq \mathcal{S}_1$. If $\mathcal{S}_2 = \emptyset$, then $I(\mathcal{S}) = \emptyset$. Otherwise, for any $\delta_{2^n}^j \in \mathcal{S}_2$, $\mathcal{S}_1$ is one step robustly reachable from $\delta_{2^n}^j$ with probability 1. If $\mathcal{S}_2 = \mathcal{S}_1$, then for any $x(t) = \delta_{2^n}^j \in \mathcal{S}_1$, we have $\mathrm{Col}_j(T_{\mathcal{S}_1|\mathcal{S}_1}) = \mathbf{1}_{2^q}$, or equivalently, $[T_{\mathcal{S}_1|\mathcal{S}_1}]_{k,j} = 1$, $\forall k \in [1 : 2^q]$. That implies that

$$\begin{aligned} 1 &= \sum_{\delta_{2^n}^i \in \mathcal{S}_1} [\mathbf{P}_k]_{i,j} \\ &= \sum_{\delta_{2^n}^i \in \mathcal{S}_1} \mathbb{P}\{x(t+1) = \delta_{2^n}^i \mid x(t) = \delta_{2^n}^j \in \mathcal{S}_1, \xi(t) = \delta_{2^q}^k\} \\ &= \mathbb{P}\{x(t+1) \in \mathcal{S}_1 \mid x(t) = \delta_{2^n}^j \in \mathcal{S}_1, \xi(t) = \delta_{2^q}^k\}, \ \forall k \in [1 : 2^q]. \end{aligned}$$

Thus $\mathcal{S}_1$ is a RIS of system (4.2) contained in $\mathcal{S}$. Hence, $\mathcal{S}_1 \subseteq I(\mathcal{S})$. On the other hand, for any $\widetilde{\delta_{2^n}^j} \in (\mathcal{S} \setminus \mathcal{S}_1)$, $\mathcal{S}$ is not one step robustly reachable from $\widetilde{\delta_{2^n}^j}$ with probability 1, i.e., $\widetilde{\delta_{2^n}^j} \notin I(\mathcal{S})$, $\forall \widetilde{\delta_{2^n}^j} \in (\mathcal{S} \setminus \mathcal{S}_1)$. That is, $I(\mathcal{S}) \bigcap (\mathcal{S} \setminus \mathcal{S}_1) = \emptyset$. Furthermore,

$$I(\mathcal{S}) \subseteq \mathcal{S}_1 \bigcup (\mathcal{S} \setminus \mathcal{S}_1).$$

Thus, $I(\mathcal{S}) \subseteq \mathcal{S}_1$, and therefore, $I(\mathcal{S}) = \mathcal{S}_1$.

Since $\mathcal{S}$ is a set with a finite number of states, there must exist an integer $\rho \in [1 : |\mathcal{S}|]$, such that $\mathcal{S}_\rho = \mathcal{S}_{\rho-1}$, where $\mathcal{S}_0 = \mathcal{S}$ and thus $I(\mathcal{S}) = \mathcal{S}_{\rho-1}$. If $\mathcal{S}_\rho = \emptyset$, then $I(\mathcal{S}) = \emptyset$.

Based on the above discussion, we propose the following algorithm to compute the LRIS with probability 1 of PBN (4.2) contained in $\mathcal{S}$.

Algorithm 4 Computation of the LRIS

Step 0: Set $\mathcal{S}_0 = \mathcal{S}$.
Step 1: For $t \geq 1$, construct the truth matrix $T_{\mathcal{S}_{t-1}|\mathcal{S}_{t-1}} \in \mathfrak{B}_{2^q \times 2^n}$:

$$[T_{\mathcal{S}_{t-1}|\mathcal{S}_{t-1}}]_{k,j} = \begin{cases} 1, \text{ if } \sum_{\delta_{2^n}^i \in \mathcal{S}_{t-1}} [\mathbf{P}_k]_{i,j} = 1, \ \forall \delta_{2^n}^j \in \mathcal{S}_{t-1}, \\ 0, \text{ otherwise.} \end{cases} \tag{4.11}$$

Compute $\mathcal{S}_t = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T_{\mathcal{S}_{t-1}|\mathcal{S}_{t-1}}) = \mathbf{1}_{2^q}\}$. If $\mathcal{S}_t = \emptyset$, then $I(\mathcal{S}) = \emptyset$, and stop. Otherwise, check whether

$$\mathcal{S}_t = \mathcal{S}_{t-1}. \tag{4.12}$$

If (4.12) holds, denote the minimum number such that (4.12) holds as $\widehat{t}$, and go to Step 2. Otherwise, let $t = t + 1$ and go to Step 1.
Step 2: The LRIS of system (4.2) contained in $\mathcal{S}$ is given by

$$I(\mathcal{S}) = \mathcal{S}_{\widehat{t}-1}. \tag{4.13}$$

In fact, from Step 1 of Algorithm 4, we immediately have the following proposition to determine whether a given set $\mathcal{S}$ is a RIS.

Proposition 4.2.1. *Given nonempty set $\mathcal{S} \subseteq \Delta_{2^n}$, the following conditions are equivalent:*

(i) *$\mathcal{S}$ is a RIS of system (4.2).*

(ii) $\mathcal{S} = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T_{\mathcal{S}|\mathcal{S}}) = \mathbf{1}_{2^q}\}$.

(iii) $\sum_{\delta_{2^n}^i \in \mathcal{S}} \sum_{k=1}^{2^q} [\mathbf{P}_k]_{i,j} = 2^q$, $\forall \delta_{2^n}^j \in \mathcal{S}$.

Proof. $(i) \Longrightarrow (ii)$ Suppose that $\mathcal{S}$ is a RIS with probability 1 of PBN (4.2). Then for any $x(t) = \delta_{2^n}^j \in \mathcal{S}$ and $\xi(t) = \delta_{2^q}^k \in \Delta_{2^q}$, it holds that

$$\begin{aligned} 1 &= \mathbb{P}\{x(t+1) \in \mathcal{S} | x(t) \in \mathcal{S}\} \\ &= \sum_{\delta_{2^n}^i \in \mathcal{S}} \mathbb{P}\{x(t+1) = \delta_{2^n}^i \mid x(t) = \delta_{2^n}^j \in \mathcal{S}\} \\ &= \sum_{\delta_{2^n}^i \in \mathcal{S}} [\mathbf{P}_k]_{i,j}, \ \forall k \in [1 : 2^q]. \end{aligned}$$

It implies that $[T_{\mathcal{S}|\mathcal{S}}]_{k,j} = 1$, $\forall k \in [1 : 2^q]$ or equivalently, $\mathrm{Col}_j(T_{\mathcal{S}|\mathcal{S}}) = \mathbf{1}_{2^q}$. Hence, (ii) holds.

$(ii) \Longrightarrow (iii)$ Since $\mathcal{S} = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T_{\mathcal{S}|\mathcal{S}}) = \mathbf{1}_{2^q}\}$, then for any $\delta_{2^n}^j \in \mathcal{S}$, we have $\mathrm{Col}_j(T_{\mathcal{S}|\mathcal{S}}) = \mathbf{1}_{2^q}$, that is $[T_{\mathcal{S}|\mathcal{S}}]_{k,j} = 1, \forall k \in [1:2^q]$. Hence, $\sum_{\delta_{2^n}^i \in \mathcal{S}} [\mathbf{P}_k]_{i,j} = 1, \forall k \in [1:2^q]$. Thus,

$$\sum_{k=1}^{2^q} \sum_{\delta_{2^n}^i \in \mathcal{S}} [\mathbf{P}_k]_{i,j} = \sum_{\delta_{2^n}^i \in \mathcal{S}} \sum_{k=1}^{2^q} [\mathbf{P}_k]_{i,j} = 2^q, \ \forall \delta_{2^n}^j \in \mathcal{S}.$$

$(iii) \Longrightarrow (i)$ From the above equation and $0 \leq \sum_{\delta_{2^n}^i \in \mathcal{S}} [\mathbf{P}_k]_{i,j} \leq 1$, we have that

$$\sum_{\delta_{2^n}^i \in \mathcal{S}} [\mathbf{P}_k]_{i,j} = 1, \forall k \in [1:2^q].$$

Then, for any $x(t) = \delta_{2^n}^j \in \mathcal{S}$ and $\xi(t) = \delta_{2^q}^k \in \Delta_{2^q}$, we have $\sum_{\delta_{2^n}^i \in \mathcal{S}} [\mathbf{P}_k]_{i,j} = \sum_{\delta_{2^n}^i \in \mathcal{S}} \mathbb{P}\{x(t+1) = \delta_{2^n}^i \mid x(t) = \delta_{2^n}^j\} = 1$. Thus, $\mathcal{S}$ is a RIS of system (4.2). □

Similarly, the following algorithm computes the LRCIS of PBCN (4.6) contained in any nonempty set $\mathcal{S} \subseteq \Delta_{2^n}$.

Algorithm 5 Computation of the LRCIS

Step 0: Set $\mathcal{S}_0 = \mathcal{S}$.
Step 1: For $t \geq 1$, construct the truth matrices $T_{\mathcal{S}_{t-1}|\mathcal{S}_{t-1}} \in \mathfrak{B}_{2^m \times 2^n}$, where

$$[T_{\mathcal{S}_{t-1}|\mathcal{S}_{t-1}}]_{l,j} = \begin{cases} 1, \ \text{if } \sum_{\delta_{2^n}^i \in \mathcal{S}_{t-1}} \sum_{k=1}^{2^q} [\overline{\mathbf{P}}_k^l]_{i,j} = 2^q, \ \forall \delta_{2^n}^j \in \mathcal{S}_{t-1}, \\ 0, \ \text{otherwise}. \end{cases} \tag{4.14}$$

Compute $\mathcal{S}_t = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T_{\mathcal{S}_{t-1}|\mathcal{S}_{t-1}}) \neq \mathbf{0}_{2^m}\}$. If $\mathcal{S}_t = \emptyset$, then $I_c(\mathcal{S}) = \emptyset$, stop the algorithm. Otherwise, check whether

$$\mathcal{S}_t = \mathcal{S}_{t-1}. \tag{4.15}$$

If (4.15) holds, denote the minimum number such that (4.15) holds as $\widehat{t}$, and go to Step 2. Otherwise, set $t = t + 1$ and go to Step 1.
Step 2: The LRCIS of system (4.6) contained in $\mathcal{S}$ is given by

$$I_c(\mathcal{S}) = \mathcal{S}_{\widehat{t}-1}. \tag{4.16}$$

Remark 4.2.1. *Since $\mathcal{S}$ is a finite set, Algorithm 5 will terminate within $\mid \mathcal{S} \mid$ steps. Moreover, from Algorithm 5, all the state feedback gain matrices $H \in \mathcal{L}_{2^m \times 2^n}$ to keep the robust control invariance of PBCN (4.6) can be determined as*

$$\begin{cases} H|_{\mathcal{S}_{\widehat{t}-1}} \leq T_{\mathcal{S}_{\widehat{t}-1}|\mathcal{S}_{\widehat{t}-1}}, \\ H|_{(\Delta_{2^n} \setminus \mathcal{S}_{\widehat{t}-1})} \leq \mathbf{1}_{2^m \times 2^n}|_{(\Delta_{2^n} \setminus \mathcal{S}_{\widehat{t}-1})}. \end{cases}$$

4.3 Finite-time robust set stability with probability 1 of PBNs

Definition 4.3.1. *(Robust set stability) The PBN (4.2) is said to be finite-time robustly stable to the nonempty set $\mathcal{S} \subseteq \Delta_{2^n}$ with probability 1, if for any initial state $x(0) \in \Delta_{2^n}$, there exists an integer $\tau \geq 0$, such that $\mathbb{P}\{x(t) \in \mathcal{S} \mid x(0) = x_0\} = 1$, $\forall t \geq \tau$ and $\{\xi(t) : t \in \mathbb{N}\} \subseteq \Delta_{2^q}$.*

Lemma 4.3.1. *The PBN (4.2) is finite-time robustly stable to $\mathcal{S}$ with probability 1, if and only if it is finite-time robustly stable to $I(\mathcal{S})$ with probability 1.*

Remark 4.3.1. *Without causing confusion, the robust set stability mentioned below is the finite-time robust set stability with probability 1.*

In the following, we consider how to determine whether system (4.2) is robustly stable to $\mathcal{S} \subseteq \Delta_{2^n}$. Firstly, compute the LRIS with probability 1 according to Algorithm 4 and assume $I(\mathcal{S}) \neq \emptyset$.

Then, let $W_1 = \Delta_{2^n} \backslash I(\mathcal{S})$ and construct the truth matrix $T_{I(\mathcal{S})|W_1} \in \mathfrak{B}_{2^q \times 2^n}$:

$$[T_{I(\mathcal{S})|W_1}]_{k,j} = \begin{cases} 1, \text{ if } \sum_{\delta_{2^n}^i \in I(\mathcal{S})} [\mathbf{P}_k]_{i,j} = 1, \ \forall \delta_{2^n}^j \in W_1, \\ 0, \text{ otherwise.} \end{cases} \tag{4.17}$$

Compute $R_1(I(\mathcal{S})) = \{\delta_{2^n}^j \mid \text{Col}_j(T_{I(\mathcal{S})|W_1}) = \mathbf{1}_{2^q}\}$ and denote $W_t = \Delta_{2^n} \backslash \overline{W}_{t-1}$, where $\overline{W}_{t-1} = \bigcup_{\lambda=0}^{t-1} R_\lambda(I(\mathcal{S}))$, $R_0(I(\mathcal{S})) = I(\mathcal{S})$, $t \geq 2$ and $t \in \mathbb{Z}_+$. Then, construct the truth matrices $T_{\overline{W}_{t-1}|W_t} \in \mathfrak{B}_{2^q \times 2^n}$ as follows:

$$[T_{\overline{W}_{t-1}|W_t}]_{k,j} = \begin{cases} 1, \text{ if } \sum_{\delta_{2^n}^i \in \overline{W}_{t-1}} [\mathbf{P}_k]_{i,j} = 1, \ \forall \delta_{2^n}^j \in W_t, \\ 0, \text{ otherwise,} \end{cases} \tag{4.18}$$

and compute $R_t(I(\mathcal{S})) = \{\delta_{2^n}^j \mid \text{Col}_j(T_{\overline{W}_{t-1}|W_t}) = \mathbf{1}_{2^q}\}$. For $t \in \mathbb{Z}_+$, it is obvious that $R_t(I(\mathcal{S})) \subseteq W_t$. Let $\overline{W}_0 = I(\mathcal{S})$ and construct the truth matrix $T_{\overline{W}_{t-1}|R_t(I(\mathcal{S}))} \in \mathfrak{B}_{2^q \times 2^n}$:

$$\text{Col}_j(T_{\overline{W}_{t-1}|R_t(I(\mathcal{S}))}) = \begin{cases} \text{Col}_j(T_{\overline{W}_{t-1}|W_t}), \text{ if } \delta_{2^n}^j \in R_t(I(\mathcal{S})), \\ \mathbf{0}_{2^q}, \text{ otherwise.} \end{cases} \tag{4.19}$$

From the computation of the robust reachable sets $R_t(I(\mathcal{S}))$, $t \in \mathbb{Z}_+$, we know that there exists at least a state in each $R_t(I(\mathcal{S}))$ and $R_i(I(\mathcal{S})) \bigcap R_j(I(\mathcal{S})) = \emptyset, \forall i, j \in \mathbb{N}, i \neq j$. If there exists an integer $\widehat{t}$ such that $R_{\widehat{t}}(I(\mathcal{S})) = \emptyset$, then $R_t(I(\mathcal{S})) = \emptyset$, $\forall t \geq \widehat{t}$. Hence, there are at most $(2^n - |I(\mathcal{S})|)$ nonempty sets. For any initial state $x(0) = \delta_{2^n}^i$, $i \in [1 : 2^n]$, if $I(\mathcal{S})$ is robustly reachable from $\delta_{2^n}^i$ with probability 1, then there must exist an integer λ, such that $\delta_{2^n}^i \in R_\lambda(I(\mathcal{S}))$ and $\lambda \leq 2^n - |I(\mathcal{S})|$.

Remark 4.3.2. *If the set $I(\mathcal{S})$ is robustly reachable with probability 1 from some initial state, then it can be robustly reachable with probability 1 within $(2^n - |I(\mathcal{S})|)$ steps.*

The evolutionary dynamics of system (4.2) can be depicted in Figure 4.1, where $t^* \in [1 : 2^n - |I(\mathcal{S})|]$.

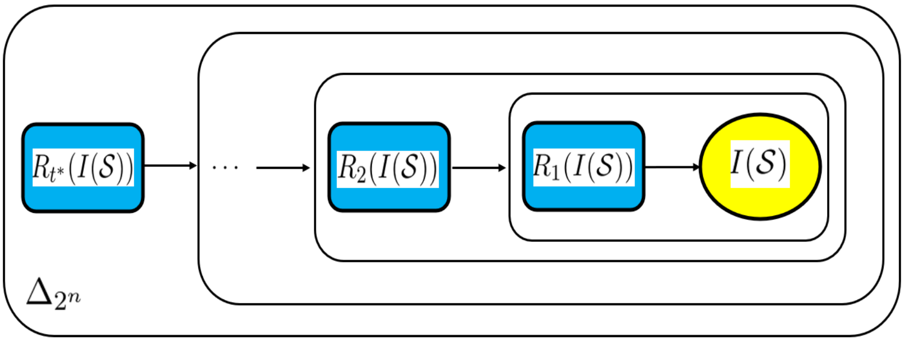

Figure 4.1: Illustration graph of evolutionary dynamics of PBN (4.2)

Based on a series of truth matrices $T_{\overline{W}_{t-1}|R_t(I(\mathcal{S}))}$, $t \in \mathbb{Z}_+$ constructed in (4.19), the following criteria are provided to detect the robust set stability of PBNs.

Theorem 4.3.2. *System (4.2) is robustly stable to $\mathcal{S} \subseteq \Delta_{2^n}$, if and only if*

(i) $I(\mathcal{S}) \neq \emptyset$,

(ii) there exists an integer $t^* \in [1 : 2^n - |I(\mathcal{S})|]$ *such that*

$$\widehat{T} = \mathbf{1}_{2^q \times 2^n},$$

where $\widehat{T} = T_{I(\mathcal{S})|I(\mathcal{S})} + \sum_{\lambda=1}^{t^*} T_{\overline{W}_{\lambda-1}|R_\lambda(I(\mathcal{S}))}$.

Proof. (Necessity) Suppose that system (4.2) is robustly stable to $\mathcal{S} \subseteq \Delta_{2^n}$, then system (4.2) is robustly stable to $I(\mathcal{S})$, which implies that $I(\mathcal{S}) \neq \emptyset$.

Since system (4.2) is robustly stable to $I(\mathcal{S})$, then there exists an integer $t^* \in [1 : 2^n - |I(\mathcal{S})|]$ such that $\bigcup_{\lambda=0}^{t^*} R_\lambda(I(\mathcal{S})) = \Delta_{2^n}$ and $R_i(I(\mathcal{S})) \bigcap R_j(I(\mathcal{S})) = \emptyset$, $\forall i, j \in [0 : t^*]$, $i \neq j$. Moreover, from Algorithm 4, we have

$$\mathrm{Col}_j(T_{I(\mathcal{S})|I(\mathcal{S})}) = \begin{cases} \mathbf{1}_{2^q}, & \text{if } \delta_{2^n}^j \in I(\mathcal{S}), \\ \mathbf{0}_{2^q}, & \text{otherwise.} \end{cases} \tag{4.20}$$

From the construction of the truth matrices $T_{\overline{W}_{\lambda-1}|W_\lambda} \in \mathfrak{B}_{2^q \times 2^n}$ and (4.19), we immediately have

$$\mathrm{Col}_j(T_{\overline{W}_{\lambda-1}|R_\lambda(I(\mathcal{S}))}) = \begin{cases} \mathbf{1}_{2^q}, & \text{if } \delta_{2^n}^j \in R_\lambda(I(\mathcal{S})), \\ \mathbf{0}_{2^q}, & \text{otherwise,} \end{cases} \tag{4.21}$$

where $\lambda \in [1 : t^*]$. Based on the above discussion, we obtain

$$\begin{aligned}\widehat{T} &= T_{I(\mathcal{S})|I(\mathcal{S})} + \sum_{\lambda=1}^{t^*} T_{\overline{W}_{\lambda-1}|R_\lambda(I(\mathcal{S}))}\\ &= \mathbf{1}_{2^q\times 2^n}.\end{aligned}$$

(Sufficiency) Assume that $I(\mathcal{S}) \neq \emptyset$ and suppose that there exists an integer $t^* \in [1 : 2^n - |I(\mathcal{S})|]$ such that $\widehat{T} = \mathbf{1}_{2^q\times 2^n}$. Let $\Lambda_0 = \{j \mid \delta_{2^n}^j \in I(\mathcal{S})\}$, $\Lambda_\lambda = \{j \mid \delta_{2^n}^j \in R_\lambda(I(\mathcal{S}))\}$, $\lambda \in [1 : t^*]$. For any $j \in [1 : 2^n]$, we have

$$\begin{aligned}\mathbf{1}_{2^q} &= \mathrm{Col}_j(\widehat{T})\\ &= \mathrm{Col}_j(T_{I(\mathcal{S})|I(\mathcal{S})}) + \sum_{\lambda=1}^{t^*} \mathrm{Col}_j(T_{\overline{W}_{\lambda-1}|R_\lambda(I(\mathcal{S}))})\\ &= \mathrm{Col}_j(T_{I(\mathcal{S})|I(\mathcal{S})}) + \mathrm{Col}_j(T_{\overline{W}_0|R_1(I(\mathcal{S}))}) +\\ &\quad \mathrm{Col}_j(T_{\overline{W}_1|R_2(I(\mathcal{S}))}) + \cdots + \mathrm{Col}_j(T_{\overline{W}_{t^*-1}|R_{t^*}(I(\mathcal{S}))}).\end{aligned}$$

Since $I(\mathcal{S}) \neq \emptyset$, from (4.20), it follows that $\forall j \in \Lambda_0$,

$$\begin{cases}\mathrm{Col}_j(T_{I(\mathcal{S})|I(\mathcal{S})}) = \mathbf{1}_{2^q},\\ \mathrm{Col}_j(T_{\overline{W}_{\lambda-1}|R_\lambda(I(\mathcal{S}))}) = \mathbf{0}_{2^q},\ \forall \lambda \in [1 : t^*].\end{cases} \tag{4.22}$$

Moreover, $R_\lambda(I(\mathcal{S})) \neq \emptyset$, $\lambda \in [1 : t^*]$. In fact, if not then $\widehat{T} < \mathbf{1}_{2^q\times 2^n}$, which is a contradiction to condition (ii). Thus, from (4.21), we obtain $\forall j \in \Lambda_{\lambda_0}$, $\lambda_0 \in [1 : t^*]$,

$$\begin{cases}\mathrm{Col}_j(T_{I(\mathcal{S})|I(\mathcal{S})}) = \mathbf{0}_{2^q},\\ \mathrm{Col}_j(T_{\overline{W}_{\lambda_0-1}|R_{\lambda_0}(I(\mathcal{S}))}) = \mathbf{1}_{2^q},\\ \mathrm{Col}_j(T_{\overline{W}_{\lambda-1}|R_\lambda(I(\mathcal{S}))}) = \mathbf{0}_{2^q},\ \forall \lambda \neq \lambda_0,\ \lambda \in [1 : t^*].\end{cases} \tag{4.23}$$

According to (4.22) and (4.23), we have $\sum_{\lambda=0}^{t^*} \mid \Lambda_\lambda \mid= 2^n$ and $\Lambda_i \bigcap \Lambda_j = \emptyset$, $\forall i, j \in [0 : t^*]$, $i \neq j$, that is, $\bigcup_{\lambda=0}^{t^*} R_\lambda(I(\mathcal{S})) = \Delta_{2^n}$ and $R_i(I(\mathcal{S})) \bigcap R_j(I(\mathcal{S})) = \emptyset$, $\forall i, j \in [0 : t^*]$, $i \neq j$. From the computation of $R_\lambda(I(\mathcal{S}))$, $\lambda \in [1 : t^*]$ and taking into account that $I(\mathcal{S})$ is a RIS with probability 1 of system (4.2), then system (4.2) is robustly stable to $I(\mathcal{S})$, which is equivalent to system (4.2) is robustly stable to $\mathcal{S}$. □

4.4 Finite-time robust set stabilization with probability 1 of PBCNs

Definition 4.4.1. *(Robust set stabilization) The PBCN (4.6) is said to be robustly stabilizable to the nonempty set $\mathcal{S} \subseteq \Delta_{2^n}$, if for any initial state $x(0) \in \Delta_{2^n}$, there is a state* *feedback control* $u(t) = Hx(t)$ *and an integer* $\tau \geq 0$, *such that* $\mathbb{P}\{x(t) \in \mathcal{S} \mid x(0) = x_0\} = 1$, $\forall t \geq \tau$ *and* $\{\xi(t) : t \in \mathbb{N}\} \subseteq \Delta_{2^q}$.

Lemma 4.4.1. *The PBCN (4.6) is robustly stabilizable to* $\mathcal{S}$, *if and only if it is robustly stabilizable to* $I_c(\mathcal{S})$.

In the following, we will study how to design $H \in \mathcal{L}_{2^m \times 2^n}$ for the robust set stabilization problem of PBCN.

For a given set $\mathcal{S} \subseteq \Delta_{2^n}$, compute its LRCIS according to Algorithm 5. Assume that $I_c(\mathcal{S}) \neq \emptyset$, then the following procedure is proposed to design all the time-optimal state feedback controls via antecedence solution technique.

Algorithm 6 Constructing the time-optimal state feedback stabilizers

Step 1: Let $\overline{W}_0 = I_c(\mathcal{S})$, $W_1 = \Delta_{2^n} \backslash \overline{W}_0$ and construct the truth matrix $T_{\overline{W}_0|W_1} \in \mathfrak{B}_{2^m \times 2^n}$:

$$[T_{\overline{W}_0|W_1}]_{l,j} = \begin{cases} 1, \text{ if } \sum_{\delta_{2^n}^i \in \overline{W}_0} \sum_{k=1}^{2^q} [\overline{\mathbf{P}}_k^l]_{i,j} = 2^q, \ \forall \delta_{2^n}^j \in W_1, \\ 0, \text{ otherwise.} \end{cases} \tag{4.24}$$

Compute $R_1(I_c(\mathcal{S})) = \{\delta_{2^n}^j \mid \text{Col}_j(T_{\overline{W}_0|W_1}) \neq \mathbf{0}_{2^m}\}$. Check whether $R_1(I_c(\mathcal{S})) \neq \emptyset$, if $R_1(I_c(\mathcal{S})) = \emptyset$, stop the algorithm. If $I_c(\mathcal{S}) \bigcup R_1(I_c(\mathcal{S})) = \Delta_{2^n}$, set $t^* = 1$ and go to Step 3; otherwise, repeat Step 2.

Step 2: For $t \geq 2$, compute $\overline{W}_{t-1} = \bigcup_{\lambda=0}^{t-1} R_\lambda(I_c(\mathcal{S}))$ and $W_t = \Delta_{2^n} \backslash \overline{W}_{t-1}$, where $R_0(I_c(\mathcal{S})) = I_c(\mathcal{S})$. Construct the truth matrix $T_{\overline{W}_{t-1}|W_t} \in \mathfrak{B}_{2^m \times 2^n}$:

$$[T_{\overline{W}_{t-1}|W_t}]_{l,j} = \begin{cases} 1, \text{ if } \sum_{\delta_{2^n}^i \in \overline{W}_{t-1}} \sum_{k=1}^{2^q} [\overline{\mathbf{P}}_k^l]_{i,j} = 2^q, \ \forall \delta_{2^n}^j \in W_t, \\ 0, \text{ otherwise.} \end{cases} \tag{4.25}$$

Compute $R_t(I_c(\mathcal{S})) = \{\delta_{2^n}^j \mid \text{Col}_j(T_{\overline{W}_{t-1}|W_t}) \neq \mathbf{0}_{2^m}\}$. If $R_t(I_c(\mathcal{S})) = \emptyset$, stop the algorithm. If

$$\bigcup_{\lambda=0}^{t} R_\lambda(I_c(\mathcal{S})) = \Delta_{2^n}, \tag{4.26}$$

denote the minimum number such that (4.26) holds as t^* and go to Step 3; otherwise, let $t = t + 1$ and go to Step 2.

Step 3: The time-optimal state feedback stabilizers $H \in \mathcal{L}_{2^m \times 2^n}$ can be constructed as follows:

$$\begin{cases} H_{|_{I_c(\mathcal{S})}} \leq T_{I_c(\mathcal{S})|I_c(\mathcal{S})}, \\ H_{|_{R_t(I_c(\mathcal{S}))}} \leq T_{\overline{W}_{t-1}|R_t(I_c(\mathcal{S}))}, \ t \in [1 : t^*], \end{cases} \tag{4.27}$$

where the truth matrix $T_{I_c(\mathcal{S})|I_c(\mathcal{S})}$ is obtained from Algorithm 5 and

$$\mathrm{Col}_j(T_{\overline{W}_{t-1}|R_t(I_c(\mathcal{S}))}) = \begin{cases} \mathrm{Col}_j(T_{\overline{W}_{t-1}|W_t}), \text{ if } \delta_{2^n}^j \in R_t(I_c(\mathcal{S})), \\ \mathbf{0}_{2^m}, \text{ otherwise.} \end{cases}$$

Remark 4.4.1. *If the set $I_c(\mathcal{S})$ is robustly reachable with probability 1 from some initial state, then there exists a state feedback control, such that it can be robustly reachable with probability 1 within $(2^n - |I_c(\mathcal{S})|)$ steps. Thus, if the robust set stabilization problem of PBCN is solvable, Algorithm 6 will terminate within $(2^n - |I_c(\mathcal{S})|)$ steps.*

Theorem 4.4.2. *The PBCN (4.6) is robustly stabilizable to $I_c(\mathcal{S})$ under the state feedback control $u(t) = Hx(t)$, if and only if there exists an integer $t^* \in [1 : 2^n - |I_c(\mathcal{S})|]$, such that*

$$\bigcup_{\lambda=0}^{t^*} R_\lambda(I_c(\mathcal{S})) = \Delta_{2^n}, \tag{4.28}$$

where $H \in \mathcal{L}_{2^m \times 2^n}$ and $R_\lambda(I_c(\mathcal{S}))$, $\lambda \in [0 : t^*]$ are obtained from Algorithm 6.

Proof. (Sufficiency) Assume that there exists an integer $t^* \in [1 : 2^n - |I_c(\mathcal{S})|]$, such that $\bigcup_{\lambda=0}^{t^*} R_\lambda(I_c(\mathcal{S})) = \Delta_{2^n}$. First, for any $\delta_{2^n}^{j_1} \in R_1(I_c(\mathcal{S}))$, there exists at least a control $H\delta_{2^n}^{j_1} = \delta_{2^m}^{l_1}$ such that

$$\sum_{\delta_{2^n}^i \in \overline{W}_0} \sum_{k=1}^{2^q} [\overline{\mathbf{P}}_k^{l_1}]_{i,j_1} = 2^q.$$

Since $0 \le \sum_{\delta_{2^n}^i \in \overline{W}_0} [\overline{\mathbf{P}}_k^{l_1}]_{i,j_1} \le 1$, then

$$\begin{aligned} 1 &= \sum_{\delta_{2^n}^i \in \overline{W}_0} [\overline{\mathbf{P}}_k^{l_1}]_{i,j_1} \\ &= \sum_{\delta_{2^n}^i \in \overline{W}_0} \mathbb{P}\{x(1) = \delta_{2^n}^i \mid x(0) = \delta_{2^n}^{j_1}, \xi(0) = \delta_{2^q}^k, u(0) = \delta_{2^m}^{l_1}\} \\ &= \mathbb{P}\{x(1) \in \overline{W}_0 \mid x(0) = \delta_{2^n}^{j_1}, \xi(0) = \delta_{2^q}^k, u(0) = \delta_{2^m}^{l_1}\}, \ \forall k \in [1 : 2^q]. \end{aligned}$$

That implies that $\delta_{2^n}^{j_1} \in R_1(I_c(\mathcal{S}))$ can reach $I_c(\mathcal{S})$ in one step with probability 1 under any disturbance $\xi \in \Delta_{2^q}$. Moreover, for any $\delta_{2^n}^{j_2} \in R_2(I_c(\mathcal{S}))$, there exists at least a control $H\delta_{2^n}^{j_2} = \delta_{2^m}^{l_2}$ such that

$$\sum_{\delta_{2^n}^i \in \overline{W}_1} \sum_{k=1}^{2^q} [\overline{\mathbf{P}}_k^{l_2}]_{i,j_2} = 2^q.$$

Then, $\delta_{2^n}^{j_2} \in R_2(I_c(\mathcal{S}))$ can reach $I_c(\mathcal{S}) \bigcup R_1(I_c(\mathcal{S}))$ in one step with probability 1 under any disturbance $\xi \in \Delta_{2^q}$. Let $\Delta_{2^q} = \Gamma_1 \bigcup \Gamma_2 \bigcup \Gamma_3$ be a partition of disturbance inputs set, where $\Gamma_i \bigcap \Gamma_j = \emptyset$, $\forall i, j \in [1 : 3]$, $i \neq j$. The reachability of $\delta_{2^n}^{j_2} \in R_2(I_c(\mathcal{S}))$ to $I_c(\mathcal{S}) \bigcup R_1(I_c(\mathcal{S}))$ implies the following cases:

(1) for any $\delta_{2^q}^{k} \in \Gamma_1$, $\delta_{2^n}^{j_2}$ can reach $R_1(I_c(\mathcal{S}))$ in one step with probability 1;

(2) for any $\delta_{2^q}^{\widetilde{k}} \in \Gamma_2$, $\delta_{2^n}^{j_2}$ can reach $I_c(\mathcal{S})$ in one step with probability 1;

(3) for any $\delta_{2^q}^{\widehat{k}} \in \Gamma_3$, $\delta_{2^n}^{j_2}$ can reach $R_1(I_c(\mathcal{S}))$ in one step with probability $p_{\widehat{k}}$ (the probability is related to the disturbance) and $\delta_{2^n}^{j_2}$ can reach $I_c(\mathcal{S})$ in one step with probability $1 - p_{\widehat{k}}$.

Note that $\Gamma_2 \neq \Delta_{2^q}$, otherwise, $\delta_{2^n}^{j_2}$ can reach $I_c(\mathcal{S})$ in one step with probability 1 under any disturbance $\xi \in \Delta_{2^q}$, it means that $\delta_{2^n}^{j_2} \in R_1(I_c(\mathcal{S}))$, which is a contradiction to $R_1(I_c(\mathcal{S})) \bigcap R_2(I_c(\mathcal{S})) = \emptyset$. No matter in which case, $I_c(\mathcal{S})$ is robustly reachable from $R_2(I_c(\mathcal{S}))$ with probability 1 after two steps.

Similarly, for any $\delta_{2^n}^{j_t} \in R_t(I_c(\mathcal{S}))$, $t \in [3 : t^*]$, there exists at least a control $H\delta_{2^n}^{j_t} = \delta_{2^m}^{l_t}$ such that $\bigcup_{\lambda=0}^{t-1} R_\lambda(I_c(\mathcal{S}))$ is robustly reachable from $R_t(I_c(\mathcal{S}))$ with probability 1. If (4.26) holds, then all states can be robustly steered to $I_c(\mathcal{S})$ with probability 1. Therefore, PBCN (4.6) is robustly stabilizable to $I_c(\mathcal{S})$.

(Necessity) We prove it by contradiction. Suppose PBCN (4.6) be robustly stabilizable to $I_c(\mathcal{S})$, but the equation (4.26) does not hold until $t = 2^n - |I_c(\mathcal{S})|$. Assume $R_{2^n - |I_c(\mathcal{S})|}(I_c(\mathcal{S})) \neq \emptyset$ and

$$\bigcup_{\lambda=0}^{2^n - |I_c(\mathcal{S})|} R_\lambda(I_c(\mathcal{S})) \neq \Delta_{2^n}.$$

It implies that there exists a state $\widehat{x} \in \Delta_{2^n} \backslash [\bigcup_{\lambda=0}^{2^n-|I_c(\mathcal{S})|} R_\lambda(I_c(\mathcal{S}))]$ such that no control can drive it to $[\bigcup_{\lambda=0}^{2^n-|I_c(\mathcal{S})|} R_\lambda(I_c(\mathcal{S}))]$ with probability 1 under the influence of disturbances in $(2^n - |I_c(\mathcal{S})|)$ steps. Hence, state $\widehat{x}$ can not reach $I_c(\mathcal{S})$ with probability 1 under any disturbance $\xi \in \Delta_{2^q}$, which contradicts the condition that the system (4.6) is robustly stabilizable to $I_c(\mathcal{S})$. □

Based on the above discussion, the following corollary is obvious.

Corollary 4.4.3. *The system (4.6) is robustly stabilizable to $\mathcal{S} \subseteq \Delta_{2^n}$ under the state feedback controller $u(t) = Hx(t)$, if and only if*

(i) $I_c(\mathcal{S}) \neq \emptyset$,

(ii) there exists an integer $t^* \in [1 : 2^n - |I_c(\mathcal{S})|]$, such that

$$\mathrm{Col}_i(\widehat{T}) \neq \mathbf{0}_{2^m}, \ \forall i \in [1 : 2^n],$$

where $\widehat{T} = T_{I_c(\mathcal{S})|I_c(\mathcal{S})} + \sum_{\lambda=1}^{t^*} T_{\overline{W}_{\lambda-1}|R_\lambda(I_c(\mathcal{S}))}$.

Moreover, if (i) and (ii) hold, then all the time-optimal state feedback gain matrices $H \in \mathcal{L}_{2^m \times 2^n}$ under which system (4.6) is robustly stabilizable to $\mathcal{S}$ can be characterized as $H \leq \widehat{T}$.

4.5 Illustrative examples

In this section, we presents two examples to demonstrate the applicability of the results obtained in this chapter.

Example 4.5.1. *Consider a disturbed PBN of the form (4.2), with $n = 3$, $q = 1$, and suppose that*

$$\begin{aligned} \mathbf{L}_1 &= \delta_8[1\ 3\ 1\ 3\ 5\ 1\ 5\ 1\ 1\ 1\ 1\ 1\ 5\ 5\ 5\ 5], \quad \mathbf{p}_1^\sigma = 0.9, \\ \mathbf{L}_2 &= \delta_8[1\ 3\ 3\ 1\ 5\ 5\ 6\ 2\ 3\ 1\ 3\ 3\ 5\ 5\ 5\ 5], \quad \mathbf{p}_2^\sigma = 0.1. \end{aligned} \tag{4.29}$$

Then, the transition probability matrix of the disturbed PBN (4.29) is given by

$$\begin{aligned} \mathbf{P} &= \mathbf{p}_1^\sigma * \mathbf{L}_1 + \mathbf{p}_2^\sigma * \mathbf{L}_2 \\ &= \begin{bmatrix} 1 & 0 & 0.9 & 0.1 & 0 & 0.9 & 0 & 0.9 & 0.9 & 1 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.1 & 0.9 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0.1 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.1 & 0.9 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\ &= [\mathbf{P}_1\ \mathbf{P}_2]. \end{aligned}$$

Let verify now whether PBN (4.29) can be robustly stable to $\mathcal{S} = \{\delta_8^1, \delta_8^3, \delta_8^5\}$.

First, according to Algorithm 4, denote $\mathcal{S}_0 = \mathcal{S}$. The truth matrix $T_{\mathcal{S}_0|\mathcal{S}_0} \in \mathfrak{B}_{2\times 8}$ is given by

$$T_{\mathcal{S}_0|\mathcal{S}_0} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{4.30}$$

From (4.30),

$$\mathcal{S}_1 = \{\delta_8^1, \delta_8^3, \delta_8^5\} = \mathcal{S}_0.$$

It is obvious that $\mathcal{S}$ is a RIS with probability 1 of system (4.29). Let now $\overline{W}_0 = \mathcal{S}$, $W_1 = \Delta_8 \backslash \overline{W}_0$ and construct the truth matrix $T_{\overline{W}_0|W_1} \in \mathfrak{B}_{2\times 8}$

$$T_{\overline{W}_0|W_1} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{4.31}$$

From (4.31), it follows that

$$R_1(\mathcal{S}) = \{\delta_8^2, \delta_8^4, \delta_8^6\}.$$

Construct now the truth matrix $T_{\overline{W}_0|R_1(\mathcal{S})} \in \mathfrak{B}_{2\times 8}$

$$T_{\overline{W}_0|R_1(\mathcal{S})} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{4.32}$$

Then, compute $\overline{W}_1 = R_0(\mathcal{S}) \bigcup R_1(\mathcal{S})$ and $W_2 = \Delta_8 \backslash \overline{W}_1$, where $R_0(\mathcal{S}) = \mathcal{S}$. Construct the truth matrix $T_{\overline{W}_1|W_2} \in \mathfrak{B}_{2\times 8}$

$$T_{\overline{W}_1|W_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \tag{4.33}$$

From (4.33), it follows that

$$R_2(\mathcal{S}) = \{\delta_8^7, \delta_8^8\}.$$

Moreover, construct the truth matrix $T_{\overline{W}_1|R_2(\mathcal{S})} = T_{\overline{W}_1|W_2}$.

Based on the above discussion, we have

$$\begin{aligned} \widehat{T} &= T_{\mathcal{S}|\mathcal{S}} + T_{\overline{W}_0|R_1(\mathcal{S})} + T_{\overline{W}_1|R_2(\mathcal{S})} \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \end{aligned}$$

It is obvious that for $t^* = 2$ we have $\widehat{T} = \mathbf{1}_{2\times 8}$. Therefore, system (4.29) is robustly stable to $\mathcal{S}$.

Example 4.5.2. *Consider the reduced disturbed PBCN model of Escherichia coli introduced in [59], which consists of the following two subnetworks:*

$$\begin{aligned} f_1 &= (X_2(t) \vee X_3(t), U(t) \wedge X_1(t), U(t) \vee (\Xi(t) \wedge X_1(t))), & \mathbf{p}_1^\sigma = 0.9, \\ f_2 &= (X_2(t) \vee X_3(t), U(t) \wedge X_1(t), X_3(t)), & \mathbf{p}_2^\sigma = 0.1, \end{aligned} \tag{4.34}$$

where the states X_1, X_2 and X_3 denote the lac mRNA, the lactose in high concentrations, and the lactose in medium concentrations, respectively; the control input U denotes the extracellular glucose and the disturbance input Ξ denotes the virus invading the Escherichia coli network.

Under the framework of algebraic formulation, we obtain

$$\begin{aligned} \overline{\mathbf{L}}_1 &= \delta_8[1\ 1\ 1\ 5\ 3\ 3\ 3\ 7\ 3\ 3\ 3\ 7\ 4\ 4\ 4\ 8\ 1\ 1\ 1\ 5\ 3\ 3\ 3\ 7\ 4\ 4\ 4\ 8\ 4\ 4\ 4\ 8], & \mathbf{p}_1^\sigma = 0.9, \\ \overline{\mathbf{L}}_2 &= \delta_8[1\ 2\ 1\ 6\ 3\ 4\ 3\ 8\ 3\ 4\ 3\ 8\ 3\ 4\ 3\ 8\ 1\ 2\ 1\ 6\ 3\ 4\ 3\ 8\ 3\ 4\ 3\ 8\ 3\ 4\ 3\ 8], & \mathbf{p}_2^\sigma = 0.1. \end{aligned}$$

Then, the control-dependent transition probability matrix of the disturbed PBCN (4.34) is given by

$$
\begin{aligned}
\overline{\mathbf{P}} &= \mathbf{p}_1^{\sigma} * \overline{\mathbf{L}}_1 + \mathbf{p}_2^{\sigma} * \overline{\mathbf{L}}_2 \\
&= \begin{bmatrix}
1 & 0.9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0.9 & 1 & 0 & 1 & 0.9 & 1 & 0 & 0.1 & 0 & 0.1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0.9 & 1 & 0.9 & 0 \\
0 & 0 & 0 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0 & 0 & 0 & 0.9 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 1 \\
1 & 0.9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0.9 & 1 & 0 & 0.1 & 0 & 0.1 & 0 & 0.1 & 0 & 0.1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0.9 & 1 & 0.9 & 0 & 0.9 & 1 & 0.9 & 0 \\
0 & 0 & 0 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix} \\
&= [\overline{\mathbf{P}}_1^1 \ \overline{\mathbf{P}}_1^2 \ \overline{\mathbf{P}}_2^1 \ \overline{\mathbf{P}}_2^2].
\end{aligned}
$$

Consider now the problem of designing all the time-optimal state feedback controllers such that system (4.34) is robustly stabilizable to the state set $\mathcal{S} = \{\delta_8^1, \delta_8^3, \delta_8^8\}$.

First, according to Algorithm 5, let $\mathcal{S}_0 = \mathcal{S}$. The truth matrix $T_{\mathcal{S}_0|\mathcal{S}_0} \in \mathfrak{B}_{2\times 8}$ is given by

$$
T_{\mathcal{S}_0|\mathcal{S}_0} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.35}
$$

From (4.35),

$$
\mathcal{S}_1 = \{\delta_8^1, \delta_8^3, \delta_8^8\} = \mathcal{S}_0.
$$

It is obvious that $\mathcal{S}$ is a RCIS with probability 1 of system (4.34). Then, according to Algorithm 6, let $\overline{W}_0 = \mathcal{S}$, $W_1 = \Delta_8 \backslash \overline{W}_0$ and construct the truth matrix $T_{\overline{W}_0|W_1} \in \mathfrak{B}_{2\times 8}$

$$
T_{\overline{W}_0|W_1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{4.36}
$$

From (4.36),

$$
R_1(\mathcal{S}) = \{\delta_8^5, \delta_8^7\}.
$$

Then, compute $\overline{W}_1 = R_0(\mathcal{S}) \bigcup R_1(\mathcal{S})$ and $W_2 = \Delta_8 \backslash \overline{W}_1$, where $R_0(\mathcal{S}) = \mathcal{S}$. Construct the truth matrix $T_{\overline{W}_1|W_2} \in \mathfrak{B}_{2\times 8}$

$$T_{\overline{W}_1|W_2} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.37}$$

From (4.37),

$$R_2(\mathcal{S}) = \{\delta_8^4\}.$$

Then, compute $\overline{W}_2 = R_0(\mathcal{S}) \bigcup R_1(\mathcal{S}) \bigcup R_2(\mathcal{S})$ and $W_3 = \Delta_8 \backslash \overline{W}_2$. Construct the truth matrix $T_{\overline{W}_2|W_3} \in \mathfrak{B}_{2\times 8}$

$$T_{\overline{W}_2|W_3} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{4.38}$$

From (4.38),

$$R_3(\mathcal{S}) = \{\delta_8^2, \delta_8^6\}.$$

Based on the above discussion, we have $T_{\overline{W}_0|R_1(\mathcal{S})} = T_{\overline{W}_0|W_1}$, $T_{\overline{W}_1|R_2(\mathcal{S})} = T_{\overline{W}_1|W_2}$ and $T_{\overline{W}_2|R_3(\mathcal{S})} = T_{\overline{W}_2|W_3}$. Thus,

$$\begin{aligned} \widehat{T} &= T_{\mathcal{S}|\mathcal{S}} + \sum_{\lambda=1}^{3} T_{\overline{W}_{\lambda-1}|R_\lambda(\mathcal{S})} \\ &= \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}. \end{aligned}$$

It is obvious that for $t^* = 3$ we have that all the columns of matrix $\widehat{T}$ are nonzero. Therefore, system (4.34) is robustly stabilizable to $\mathcal{S} = \{\delta_8^1, \delta_8^3, \delta_8^8\}$ under the state feedback control $u(t) = Hx(t)$. Moreover, there are 4 choices of time-optimal state feedback gain matrices $H \in \mathcal{L}_{2\times 8}$:

$$\begin{aligned} H &= \delta_2[1\ 2\ 1\ 1\ 2\ 1\ 1\ 1], \\ H &= \delta_2[1\ 2\ 1\ 1\ 2\ 1\ 1\ 2], \\ H &= \delta_2[1\ 2\ 1\ 1\ 2\ 2\ 1\ 1], \\ H &= \delta_2[1\ 2\ 1\ 1\ 2\ 2\ 1\ 2]. \end{aligned}$$

4.6 Conclusions

In this chapter, the finite-time robust set stability with probability 1 of PBNs and finite-time robust set stabilization with probability 1 of PBCNs, have been investigated respectively. On one hand, we proposed an algorithm to compute the LRIS with probability 1 contained in a given state set and presented necessary and sufficient conditions to determine the finite-time robust set stability with probability 1 of PBNs. On the other hand, we have constructed an algorithm to compute the LRCIS with probability 1 contained in a given state set. Based on this RCIS, we derived some necessary and sufficient conditions to detect whether the PBCNs are finite-time robustly stabilized to the given set with probability 1. Furthermore, we have shown that all the time-optimal controllers can be obtained via antecedence solution technique. Illustrative examples have also been given to show the effectiveness of the main results of this chapter.

Chapter 5

Stabilization and Set Stabilization of Periodic Switched Boolean Control Networks

5.1 Introduction

It is well known that stabilizability analysis plays one of the most basic and important roles in control theory. A typical example is therapeutic interventions, that is, driving the GRN to a healthy state, and maintain this state forever [74]. In other cases, it is necessary to study whether the system can be globally stabilized to a given state set instead of a single point, which is known as set stabilization. However, until now, there have been few results on stabilization or set stabilization of SBCNs. For instance, under arbitrary switching signal, Yerudkar et al. investigated the design of switching-signal-dependent state feedback and output feedback controllers for the stabilization of SBCNs [130]. Li et al. presented necessary and sufficient conditions for set stabilization of SBCNs under arbitrary switching signal for the case of switching-signal-dependent controller or switching-signal-independent controller [59], and pointed that "the condition of switching-signal-dependent controller is less conservative than the one of the switching-signal-independent controller." To the best of our knowledge, the problem of stabilization or set stabilization analysis for SBCNs under periodic switching signal has not been addressed before.

This chapter investigates the stabilization and set stabilization problems of SBCNs with periodic switching signal, and both open loop stabilizers and state feedback stabilizers are designed if the problems are solvable. The main contributions of this chapter are:

- Necessary and sufficient conditions for the global stabilization and global set stabilization are derived.
- A constructive procedure is proposed to design open loop controller for the stabilization problem.
- Algorithms based on antecedence solution technique are established to design switching-signal-dependent state feedback controllers for the stabilization and set stabilization problems.

5.2 Problem formulation

The dynamics of periodic SBCNs with n nodes, m control inputs and a periodic switching signal with ω values can be described as follows:

$$\begin{cases} X_1(t+1) = f_1^{\sigma(t)}(X(t); U(t)), \\ X_2(t+1) = f_2^{\sigma(t)}(X(t); U(t)), \\ \qquad \vdots \\ X_n(t+1) = f_n^{\sigma(t)}(X(t); U(t)), \end{cases} \tag{5.1}$$

where $X(t) = (X_1(t), X_2(t), \cdots, X_n(t)) \in \mathcal{D}^n$ and $U(t) = (U_1(t), U_2(t), \cdots, U_m(t)) \in \mathcal{D}^m$ are the state and control, respectively. Here, $X_i \in \mathcal{D}, i \in [1 : n]$, $U_j \in \mathcal{D}, j \in [1 : m]$ are logical variables, and $f_i^{\sigma(t)} : \mathcal{D}^{n+m} \longmapsto \mathcal{D}$, $i \in [1 : n]$ is a logical function. Moreover, the periodic switching law with period ω has the following form:

$$\sigma(t) = \begin{cases} 1, & t \ mod\ \omega = 0, \\ 2, & t \ mod\ \omega = 1, \\ \vdots \\ \omega, & t \ mod\ \omega = \omega - 1. \end{cases} \tag{5.2}$$

Remark 5.2.1. *A periodic switching signal with ω values may have $\omega!$ different expressions, that is, all the possible permutations of $\{1, 2, \cdots, \omega\}$. Here we just consider one of the possible periodic switching forms, and all the results obtained in this chapter can be generalized to any other periodic switching expressions.*

Let x_i and u^j be the vector form of X_i and U_j respectively. Based on Lemma 1.2.1, for any logical function $f_i^{\sigma(t)}$, $i \in [1:n]$, there exists a unique structural matrix $M_i^{\sigma(t)} \in \mathcal{L}_{2\times 2^{n+m}}$ such that system (5.1) can be converted into

$$x_i(t+1) = M_i^{\sigma(t)} u(t)x(t), \quad i \in [1:n], \tag{5.3}$$

where $u(t) = \ltimes_{j=1}^m u^j(t) \in \Delta_{2^m}$ and $x(t) = \ltimes_{i=1}^n x_i(t) \in \Delta_{2^n}$. Then, multiplying all the equations yields

$$x(t+1) = L_{\sigma(t)} u(t)x(t), \tag{5.4}$$

where $L_{\sigma(t)} = M_1^{\sigma(t)} * M_2^{\sigma(t)} * \cdots * M_n^{\sigma(t)} \in \mathcal{L}_{2^n\times 2^{n+m}}$. Furthermore, let $L = [L_1\ L_2\ \cdots\ L_\omega] \in \mathcal{L}_{2^n\times\omega\cdot 2^{n+m}}$ and $i \sim \delta_\omega^i, i \in [1:\omega]$, then when $\sigma(t) = \delta_\omega^i$, we have $L\sigma(t) = L_i$. Thus, the algebraic formulation of system (5.1) is given by

$$x(t+1) = L\sigma(t)u(t)x(t). \tag{5.5}$$

In this chapter, two kinds of controls are considered:

(i) Open-loop controller: the control is a free Boolean sequence, that is, the control is a designed sequence $U(0), U(1), \cdots$.

(ii) Switching-signal-dependent state feedback controller: the controls are expressed by state variables satisfying certain logical rule under periodic switching signal, such as

$$\begin{cases} U_{1,\sigma(t)}(t) = h_{1,\sigma(t)}(X_1(t), X_2(t), \cdots, X_n(t)), \\ U_{2,\sigma(t)}(t) = h_{2,\sigma(t)}(X_1(t), X_2(t), \cdots, X_n(t)), \\ \vdots \\ U_{m,\sigma(t)}(t) = h_{m,\sigma(t)}(X_1(t), X_2(t), \cdots, X_n(t)), \end{cases} \tag{5.6}$$

whose algebraic form can be expressed as

$$u_{\sigma(t)}(t) = H_{\sigma(t)}x(t), \tag{5.7}$$

where $H_{\sigma(t)} \in \mathcal{L}_{2^m\times 2^n}$ is the switching-signal-dependent state feedback gain matrix.

Definition 5.2.1. *(Stabilization) The SBCN (5.1) is said to be globally stabilizable (or feedback stabilizable) to a given state $x^* \in \Delta_{2^n}$ under periodic switching law (5.2), if for any initial state $x(0) \in \Delta_{2^n}$, there exists a control sequence $u(0), u(1), \cdots$ (or a feedback control law $u_{\sigma(t)}(t) = H_{\sigma(t)}x(t)$) and an integer $T \geq 0$, such that $x(t) = x^*$, for every $t \geq T$.*

Definition 5.2.2. *(Set stabilization) The SBCN (5.1) is said to be stabilizable (or feedback stabilizable) to the nonempty set* $\Gamma \subseteq \Delta_{2^n}$ *under periodic switched law (5.2), if for any initial state* $x(0) \in \Delta_{2^n}$*, there exists a control sequence* $u(0), u(1), \cdots$ *(or a feedback control law* $u_{\sigma(t)}(t) = H_{\sigma(t)}x(t)$*) and an integer* $T \geq 0$*, such that* $x(t) \in \Gamma$ *holds, for every* $t \geq T$*.*

Definition 5.2.3. *(Common control fixed point) A state* $x^* = \delta_{2^n}^{\gamma} \in \Delta_{2^n}$ *is said to be the common control fixed point of (5.5), if for any sub-system* $x(t+1) = L_\nu u(t)x(t)$*, there exists a control* $u_\nu \in \Delta_{2^m}$ *(the control is related to the subsystem), such that* $L_\nu u_\nu x^* = x^*$*.*

Furthermore, define

$$\begin{aligned}
\mathcal{C}_1^* &= \{\delta_{2^m}^{k_1} | L_1 \delta_{2^m}^{k_1} \delta_{2^n}^{\gamma} = \delta_{2^n}^{\gamma}, k_1 \in [1:2^m]\},\\
\mathcal{C}_2^* &= \{\delta_{2^m}^{k_2} | L_2 \delta_{2^m}^{k_2} \delta_{2^n}^{\gamma} = \delta_{2^n}^{\gamma}, k_2 \in [1:2^m]\},\\
&\vdots\\
\mathcal{C}_\omega^* &= \{\delta_{2^m}^{k_\omega} | L_\omega \delta_{2^m}^{k_\omega} \delta_{2^n}^{\gamma} = \delta_{2^n}^{\gamma}, k_\omega \in [1:2^m]\}.
\end{aligned}$$

Lemma 5.2.1. *The state* $x^* = \delta_{2^n}^{\gamma}$ *is a common control fixed point if and only if* $\mathcal{C}_\nu^* \neq \emptyset$*, for any* $\nu \in [1:\omega]$*. In particular, the state* $x^* = \delta_{2^n}^{\gamma}$ *is a common control fixed point under common control if and only if* $\bigcap_{\nu=1}^{\omega} \mathcal{C}_\nu^* \neq \emptyset$*.*

It is obvious that if $x^* = \delta_{2^n}^{\gamma}$ is a common control fixed point, then for any sub-system, there exists at least a control u_ν such that $L_\nu u_\nu x^* = x^*$, $\forall \nu \in [1:\omega]$. Thus, $u_\nu \in \mathcal{C}_\nu^*$. Conversely, if $\mathcal{C}_\nu^* \neq \emptyset$, $\forall \nu \in [1:\omega]$, then $x^* = \delta_{2^n}^{\gamma}$ is a control fixed point of each sub-system.

Definition 5.2.4. *(Common control invariant set) A nonempty set* $\Gamma \subseteq \Delta_{2^n}$ *is said to be the common control invariant set of (5.5), if for any sub-system* $x(t+1) = L_\nu u(t)x(t)$ *and any* $x(t) = \delta_{2^n}^{j} \in \Gamma$*, there exists at least a control* $u_{(\nu,j)} \in \Delta_{2^m}$ *(the control is related to the subsystem and the state in* Γ*), such that* $L_\nu u_{(\nu,j)} \delta_{2^n}^{j} \in \Gamma$*.*

Let now

$$\begin{aligned}
\mathcal{C}_{1,j}^{\Gamma} &= \{\delta_{2^m}^{k_{(1,j)}} | L_1 \delta_{2^m}^{k_{(1,j)}} \delta_{2^n}^{j} \in \Gamma,\ if\ \delta_{2^n}^{j} \in \Gamma\},\\
\mathcal{C}_{2,j}^{\Gamma} &= \{\delta_{2^m}^{k_{(2,j)}} | L_2 \delta_{2^m}^{k_{(2,j)}} \delta_{2^n}^{j} \in \Gamma,\ if\ \delta_{2^n}^{j} \in \Gamma\},\\
&\vdots\\
\mathcal{C}_{\omega,j}^{\Gamma} &= \{\delta_{2^m}^{k_{(\omega,j)}} | L_\omega \delta_{2^m}^{k_{(\omega,j)}} \delta_{2^n}^{j} \in \Gamma,\ if\ \delta_{2^n}^{j} \in \Gamma\}.
\end{aligned}$$

The following lemma is proposed to detect whether a given set is a common control invariant set of system (5.5).

Lemma 5.2.2. *The nonempty set* $\Gamma \subseteq \Delta_{2^n}$ *is a common control invariant set if and only if* $\mathcal{C}_{\nu,j}^{\Gamma} \neq \emptyset$*, for any* $\nu \in [1:\omega]$*, for all* $\delta_{2^n}^{j} \in \Gamma$*. Particularly, the nonempty set* $\Gamma \subseteq \Delta_{2^n}$ *is a common control invariant set under common control if and only if* $\bigcap_{\nu=1}^{\omega} \mathcal{C}_{\nu,j}^{\Gamma} \neq \emptyset$*,* $\forall \delta_{2^n}^{j} \in \Gamma$*.*

5.3 Controller design for stabilization of periodic SBCNs

In this section, we study whether the SBCNs with periodic switching signal can be stabilized by open loop controller and state feedback controller respectively, and present the constructive procedures of open loop controller as well as the design algorithms of switching-signal-dependent state feedback controller via antecedence solution technique.

Theorem 5.3.1. *Consider SBCNs (5.5) with periodic switching signal. The system is globally stabilizable to a common control fixed point* $x^* = \delta_{2^n}^{\gamma}$ *under a free-type control sequence if and only if there exist an integer* $T^* = p\omega + q + 1$ *and* $\alpha \in [1 : 2^{T^* m}]$*, such that*

$$\mathrm{Blk}_\alpha(\widetilde{L}) = \delta_{2^n}[\underbrace{\gamma\ \gamma\ \cdots\ \gamma}_{2^n}], \tag{5.8}$$

where $p \geq 0$*,* $q \in [0 : \omega-1]$ *and* $\widetilde{L} = [L_{q+1}(I_{2^m} \otimes L_q)(I_{2^{2m}} \otimes L_{q-1}) \cdots (I_{2^{qm}} \otimes L_1)(I_{2^{(q+1)m}} \otimes L_\omega)$ $(I_{2^{(q+2)m}} \otimes L_{\omega-1}) \cdots (I_{2^{(\omega+q)m}} \otimes L_1)(I_{2^{(\omega+q+1)m}} \otimes L_\omega)(I_{2^{(\omega+q+2)m}} \otimes L_{\omega-1}) \cdots (I_{2^{(2\omega+q)m}} \otimes L_1) \cdots$ $(I_{2^{((p-1)\omega+q+1)m}} \otimes L_\omega)(I_{2^{((p-1)\omega+q+2)m}} \otimes L_{\omega-1}) \cdots (I_{2^{(p\omega+q)m}} \otimes L_1)] \in \mathcal{L}_{2^n \times 2^{n+T^* m}}$*. In addition, if (5.8) holds, then the free-type control sequence which makes the system globally stabilized is given by*

$$u(t) = \begin{cases} \widetilde{u}(t), \ 0 \leq t \leq T^* - 1, \\ \delta_{2^m}^{k_1} \in \mathcal{C}_1^*, \ t \geq T^* \text{ and } t \bmod \omega = 0, \\ \delta_{2^m}^{k_2} \in \mathcal{C}_2^*, \ t \geq T^* \text{ and } t \bmod \omega = 1, \\ \vdots \\ \delta_{2^m}^{k_\omega} \in \mathcal{C}_\omega^*, \ t \geq T^* \text{ and } t \bmod \omega = \omega - 1, \end{cases} \tag{5.9}$$

where $\widetilde{u}(T^*-1)\widetilde{u}(T^*-2)\cdots\widetilde{u}(0) = \delta_{2^{T^* m}}^{\alpha}$*.*

Proof. For any $t \in \mathbb{Z}_+$, the dynamics of system (5.5) can be expressed as

$$\begin{aligned} x(1) &= L\sigma(0)u(0)x(0) \\ &= L\delta_\omega^1 u(0)x(0) \\ &= L_1 u(0)x(0), \\ x(2) &= L\sigma(1)u(1)x(1) \\ &= L\delta_\omega^2 u(1)L_1 u(0)x(0) \\ &= L_2(I_{2^m} \otimes L_1)u(1)u(0)x(0), \\ &\vdots \end{aligned}$$

$$
\begin{aligned}
x(\omega-1) &= L\sigma(\omega-2)u(\omega-2)x(\omega-2)\\
&= L\sigma(\omega-2)u(\omega-2)L\sigma(\omega-3)u(\omega-3)\cdots L\sigma(1)u(1)L\sigma(0)u(0)x(0)\\
&= L\delta_{\omega}^{\omega-1}u(\omega-2)L_{\omega-2}(I_{2^m}\otimes L_{\omega-3})(I_{2^{2m}}\otimes L_{\omega-4})\cdots(I_{2^{(\omega-3)m}}\otimes L_1)u(\omega-3)\cdots\\
&\quad u(0)x(0)\\
&= L_{\omega-1}(I_{2^m}\otimes L_{\omega-2})u(\omega-2)(I_{2^m}\otimes L_{\omega-3})(I_{2^{2m}}\otimes L_{\omega-4})\cdots(I_{2^{(\omega-3)m}}\otimes L_1)\\
&\quad u(\omega-3)\cdots u(0)x(0)\\
&\cdots\\
&= L_{\omega-1}(I_{2^m}\otimes L_{\omega-2})(I_{2^{2m}}\otimes L_{\omega-3})\cdots(I_{2^{(\omega-2)m}}\otimes L_1)u(\omega-2)u(\omega-3)\cdots\\
&\quad u(0)x(0),\\
x(\omega) &= L\sigma(\omega-1)u(\omega-1)x(\omega-1)\\
&= L\delta_{\omega}^{\omega}u(\omega-1)L_{\omega-1}(I_{2^m}\otimes L_{\omega-2})(I_{2^{2m}}\otimes L_{\omega-3})\cdots(I_{2^{(\omega-2)m}}\otimes L_1)u(\omega-2)\\
&\quad u(\omega-3)\cdots u(0)x(0)\\
&= L_{\omega}(I_{2^m}\otimes L_{\omega-1})u(\omega-1)(I_{2^m}\otimes L_{\omega-2})(I_{2^{2m}}\otimes L_{\omega-3})\cdots(I_{2^{(\omega-2)m}}\otimes L_1)\\
&\quad u(\omega-3)\cdots u(0)x(0)\\
&\cdots\\
&= L_{\omega}(I_{2^m}\otimes L_{\omega-1})(I_{2^{2m}}\otimes L_{\omega-2})\cdots(I_{2^{(\omega-1)m}}\otimes L_1)u(\omega-1)u(\omega-2)\cdots u(0)\\
&\quad x(0),\\
&\vdots\\
x(t) &= L\sigma(t-1)u(t-1)x(t-1)\\
&= L_{q+1}(I_{2^m}\otimes L_q)(I_{2^{2m}}\otimes L_{q-1})\cdots(I_{2^{qm}}\otimes L_1)(I_{2^{(q+1)m}}\otimes L_{\omega})(I_{2^{(q+2)m}}\otimes L_{\omega-1})\cdots\\
&\quad (I_{2^{(\omega+q)m}}\otimes L_1)(I_{2^{(\omega+q+1)m}}\otimes L_{\omega})(I_{2^{(\omega+q+2)m}}\otimes L_{\omega-1})\cdots(I_{2^{(2\omega+q)m}}\otimes L_1)\cdots\\
&\quad (I_{2^{((p-1)\omega+q+1)m}}\otimes L_{\omega})(I_{2^{((p-1)\omega+q+2)m}}\otimes L_{\omega-1})\cdots(I_{2^{(p\omega+q)m}}\otimes L_1)u(t-1)\cdots u(0)\\
&\quad x(0)\\
&:= \tilde{L}u(t-1)u(t-2)\cdots u(0)x(0),
\end{aligned}
$$

where $t=p\omega+q+1$, $p\geq 0$, $q\in[0:\omega-1]$ and $\widetilde{L}=[L_{q+1}(I_{2^m}\otimes L_q)(I_{2^{2m}}\otimes L_{q-1})\cdots(I_{2^{qm}}\otimes L_1)(I_{2^{(q+1)m}}\otimes L_{\omega})(I_{2^{(q+2)m}}\otimes L_{\omega-1})\cdots(I_{2^{(\omega+q)m}}\otimes L_1)(I_{2^{(\omega+q+1)m}}\otimes L_{\omega})(I_{2^{(\omega+q+2)m}}\otimes L_{\omega-1})\cdots(I_{2^{(2\omega+q)m}}\otimes L_1)\cdots(I_{2^{((p-1)\omega+q+1)m}}\otimes L_{\omega})(I_{2^{((p-1)\omega+q+2)m}}\otimes L_{\omega-1})\cdots(I_{2^{(p\omega+q)m}}\otimes L_1)]\in\mathcal{L}_{2^n\times 2^{n+tm}}$.

(Necessity) Assume that the dynamics of system (5.5) are globally stabilizable to $x^*=\delta_{2^n}^{\gamma}$. Then, for any initial state $x(0)\in\Delta_{2^n}$, the iterative sequence will converge to $\delta_{2^n}^{\gamma}$. Then there must exist an integer $T^*=p\omega+q+1$, and a sequence of control $u(0),u(1),\cdots,u(T^*-$

1), where $u(T^*-1)u(T^*-2)\cdots u(0)=\delta_{2^{T^*m}}^{\alpha}$, such that

$$\begin{aligned} x(T^*) &= \tilde{L}u(T^*-1)u(T^*-2)\cdots u(0)x(0) \\ &= \mathrm{Blk}_\alpha(\tilde{L})\delta_{2^n}^i \\ &= \delta_{2^n}^\gamma,\ \forall i\in[1:2^n], \end{aligned}$$

which implies that (5.8) is satisfied. Moreover, for $t\geq T^*$, it holds that

$$x(t+1)=x^*.$$

Thus, $x^*=L\sigma(t)u(t)x^*$ for all $t\geq T^*$, then (5.9) must hold.

(Sufficiency) Suppose that (5.8) is satisfied. We prove that the system (5.5) is globally convergent to x^* under the free-type control sequence (5.9). From (5.8), one can obtain that for any $x(0)\in\Delta_{2^n}$,

$$x(T^*)=Blk_\alpha(\tilde{L})x(0)=\delta_{2^n}[\gamma\ \gamma\ \cdots\ \gamma]x(0)=\delta_{2^n}^\gamma.$$

Since $x^*=\delta_{2^n}^\gamma$ is a common control fixed point of system (5.5), hence the system (5.5) globally converges to $x^*=\delta_{2^n}^\gamma$. □

In the following, we provide an illustrative example to show how to design an open loop controller such that the SBCNs with periodic switching signal can globally stabilize to a common control fixed point.

Example 5.3.1. *Consider the following reduced Boolean model for the lactose operon in the bacterium Escherichia coli, which is presented in [116].*

$$\begin{cases} X_1(t+1)=\neg U_1(t)\wedge(X_2(t)\vee X_3(t)), \\ X_2(t+1)=\neg U_1(t)\wedge U_2(t)\wedge X_1(t), \\ X_3(t+1)=\neg U_1(t)\wedge(U_2(t)\vee(U_3(t)\wedge X_1(t))), \end{cases} \tag{5.10}$$

where X_1, X_2 and X_3 are states which denote the lac mRNA, the lactose in high concentrations, and the lactose in medium concentrations, respectively; U_1, U_2 and U_3 are controls which represent the extracellular glucose, the high extracellular lactose, and the medium extracellular lactose, respectively.

Fix $U_1(t)\equiv 0$ and let $V_1(t)=U_2(t)$, $V_2(t)=U_3(t)$; then system (5.10) can be converted to

$$\begin{cases} X_1(t+1)=X_2(t)\vee X_3(t), \\ X_2(t+1)=V_1(t)\wedge X_1(t), \\ X_3(t+1)=V_1(t)\vee(V_2(t)\wedge X_1(t)). \end{cases} \tag{5.11}$$

Assuming that the state X_3 in system (5.11) does not update its value, then system (5.11) can be expressed as

$$\begin{cases} X_1(t+1) = X_2(t) \vee X_3(t), \\ X_2(t+1) = V_1(t) \wedge X_1(t), \\ X_3(t+1) = X_3(t). \end{cases} \tag{5.12}$$

If system (5.11) and (5.12) are sub-systems of a SBCN, then the SBCN can be epressed as:

$$\begin{cases} X_1(t+1) = f_1^{\sigma(t)}(X_1(t), X_2(t), X_3(t), V_1(t), V_2(t)), \\ X_2(t+1) = f_2^{\sigma(t)}(X_1(t), X_2(t), X_3(t), V_1(t), V_2(t)), \\ X_3(t+1) = f_3^{\sigma(t)}(X_1(t), X_2(t), X_3(t), V_1(t), V_2(t)), \end{cases} \tag{5.13}$$

where $f_1^1 = f_1^2 = X_2(t) \vee X_3(t)$, $f_2^1 = f_2^2 = V_1(t) \wedge X_1(t)$, $f_3^1 = V_1(t) \vee (V_2(t) \wedge X_1(t))$, and $f_3^2 = X_3(t)$. Assume the periodic switching law has the following form:

$$\sigma(t) = \begin{cases} 1, & t \ mod \ 2 = 0, \\ 2, & t \ mod \ 2 = 1. \end{cases} \tag{5.14}$$

Denoting $x(t) = \ltimes_{i=1}^3 x_i(t) \in \Delta_8$, $v(t) = \ltimes_{j=1}^2 v_j(t) \in \Delta_4$ and identifying the switching signal $1 \sim \delta_2^1$, $2 \sim \delta_2^2$, the algebraic form of system (5.13) can be expressed as:

$$x(t+1) = L\sigma(t)v(t)x(t),$$

where $L = [L_1 \ L_2]$, and

$$\begin{aligned} L_1 &= \delta_8[1\ 1\ 1\ 5\ 3\ 3\ 3\ 7\ 1\ 1\ 1\ 5\ 3\ 3\ 3\ 7\ 3\ 3\ 3\ 7\ 4\ 4\ 4\ 8\ 3\ 3\ 3\ 7\ 4\ 4\ 4\ 8], \\ L_2 &= \delta_8[1\ 2\ 1\ 6\ 3\ 4\ 3\ 8\ 1\ 2\ 1\ 6\ 3\ 4\ 3\ 8\ 3\ 4\ 3\ 8\ 3\ 4\ 3\ 8\ 3\ 4\ 3\ 8\ 3\ 4\ 3\ 8]. \end{aligned}$$

Assume that the equilibrium $x^* = \delta_8^3$, which represents the lactose operon being on. We are now ready to design a free-type control sequence to stabilize system (5.13) to $x^* = \delta_8^3$.

First, we have

$$\begin{aligned} \mathcal{C}_1^* &= \{\delta_4^{k_1} | L_1 \delta_4^{k_1} \delta_8^3 = \delta_8^3, 1 \le k_1 \le 4\} = \{\delta_4^3, \delta_4^4\}, \\ \mathcal{C}_2^* &= \{\delta_4^{k_2} | L_2 \delta_4^{k_2} \delta_8^3 = \delta_8^3, 1 \le k_2 \le 4\} = \{\delta_4^3, \delta_4^4\}. \end{aligned}$$

A straightforward calculation shows that $T^* = 2$, $\alpha = 9$, and

$$\text{Blk}_9(\widetilde{L}) = \delta_8[3\ 3\ 3\ 3\ 3\ 3\ 3\ 3],$$

where $\widetilde{L} = L_2(I_{2^2} \otimes L_1) \in \mathcal{L}_{2^3 \times 2^7}$. Since $\delta_4^3 \times \delta_4^1 = \delta_{16}^9$, from (5.9), one can design the free-type control sequence as follows

$$u(t) = \begin{cases} \delta_4^1, \ t = 0, \\ \delta_4^3, \ t = 1, \\ \delta_4^{k_1} \in \mathcal{C}_1^*, \ t \geq 2 \ and \ t \ mod \ 2 = 0, \\ \delta_4^{k_2} \in \mathcal{C}_2^*, \ t \geq 2 \ and \ t \ mod \ 2 = 1. \end{cases} \tag{5.15}$$

In the following, necessary and sufficient conditions to determine whether the SBCNs with periodic switching signal can be stabilized under switching-signal-dependent state feedback controllers are investigated. Moreover, the constructive algorithm of state feedback controllers is presented.

Remark 5.3.1. *If the state $x^* = \delta_{2^n}^\gamma$ is reachable from some initial state, then there exists a control sequence such that it can be reachable within $\omega \cdot (2^n - 1)$ steps.*

The switching-signal-dependent logical matrices $H_{\sigma(t)} \in \mathcal{L}_{2^m \times 2^n}$ which can stabilize the system (5.5) to $x^* = \delta_{2^n}^\gamma$ can be constructed as follows.

Algorithm 7 Constructing switching-signal-dependent state feedback stabilizers

Step 1: Let $R_0(x^*) = \{\delta_{2^n}^\gamma\}$, and construct the truth matrices $T^\nu_{R_0(x^*)} \in \mathfrak{B}_{2^m \times 2^n}$, $\nu \in [1:\omega]$ as:

$$[T^\nu_{R_0(x^*)}]_{i,j} = \begin{cases} 1, \ \text{if } L_\nu \delta_{2^m}^i \delta_{2^n}^j = \delta_{2^n}^\gamma, \\ 0, \ \text{otherwise}. \end{cases} \tag{5.16}$$

Compute $R_1^\nu(x^*) = \{\delta_{2^n}^j \mid \text{Col}_j(T^\nu_{R_0(x^*)}) \neq \mathbf{0}_{2^m}\}$, $\nu \in [1:\omega]$ and check whether $x^* \in \bigcap_{\nu=1}^{\omega} R_1^\nu(x^*)$. If $x^* \notin \bigcap_{\nu=1}^{\omega} R_1^\nu(x^*)$, then x^* is not a common control fixed point, and the construction problem of stabilizer is not solvable, stop the algorithm. If $R_1^1(x^*) = \Delta_{2^n}$, denote $T^* = 1$ and go to Step 5. Otherwise, go to Step 2.

Step 2: Construct the truth matrix $T^1_{R_1^2(x^*)} \in \mathfrak{B}_{2^m \times 2^n}$:

$$[T^1_{R_1^2(x^*)}]_{i,j} = \begin{cases} 1, \ \text{if } L_1 \delta_{2^m}^i \delta_{2^n}^j \in R_1^2(x^*), \forall \delta_{2^n}^j \in \Delta_{2^n} \backslash R_1^1(x^*), \\ 0, \ \text{otherwise}. \end{cases} \tag{5.17}$$

Compute $R_2^1(x^*) = \{\delta_{2^n}^j \mid \text{Col}_j(T^1_{R_1^2(x^*)}) \neq \mathbf{0}_{2^m}\}$. If $R_2^1(x^*) \neq \emptyset$ and $R_1^1(x^*) \bigcup R_2^1(x^*) = \Delta_{2^n}$, denote $T^* = 2$ and go to Step 5. Otherwise, go to Step 3.

Step 3: For $t \in [3:\omega]$, construct the truth matrix $T^{t-1}_{R_1^t(x^*)} \in \mathfrak{B}_{2^m \times 2^n}$:

$$[T^{t-1}_{R_1^t(x^*)}]_{i,j} = \begin{cases} 1, \ \text{if } L_{t-1} \delta_{2^m}^i \delta_{2^n}^j \in R_1^t(x^*), \\ 0, \ \text{otherwise}. \end{cases} \tag{5.18}$$

Compute $R_2^{t-1}(x^*) = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T^{t-1}_{R_1^t(x^*)}) \neq \mathbf{0}_{2^m}\}$ and construct the truth matrix $T^{t-2}_{R_2^{t-1}(x^*)} \in \mathfrak{B}_{2^m\times 2^n}$:

$$[T^{t-2}_{R_2^{t-1}(x^*)}]_{i,j} = \begin{cases} 1, \text{ if } L_{t-2}\delta_{2^m}^i\delta_{2^n}^j \in R_2^{t-1}(x^*), \\ 0, \text{ otherwise.} \end{cases} \tag{5.19}$$

Compute $R_3^{t-2}(x^*) = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T^{t-2}_{R_2^{t-1}(x^*)}) \neq \mathbf{0}_{2^m}\}$. Repeat this procedure, constructing truth matrices $T^{t-3}_{R_3^{t-2}(x^*)}$, $T^{t-4}_{R_4^{t-3}(x^*)}, \cdots, T^{2}_{R_{t-2}^{3}(x^*)}$ respectively, and compute $R_4^{t-3}(x^*)$, $R_5^{t-4}(x^*)$, $\cdots$, $R_{t-1}^2(x^*)$.
Then, construct the truth matrix $T^1_{R_{t-1}^2(x^*)} \in \mathfrak{B}_{2^m\times 2^n}$:

$$[T^1_{R_{t-1}^2(x^*)}]_{i,j} = \begin{cases} 1, \text{ if } L_1\delta_{2^m}^i\delta_{2^n}^j \in R_{t-1}^2(x^*), \forall \delta_{2^n}^j \in \Delta_{2^n}\backslash(\bigcup_{k=1}^{t-1} R_k^1(x^*)), \\ 0, \text{ otherwise.} \end{cases} \tag{5.20}$$

Compute $R_t^1(x^*) = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T^1_{R_{t-1}^2(x^*)}) \neq \mathbf{0}_{2^m}\}$. If $R_t^1(x^*) \neq \emptyset$ and

$$\bigcup_{k=1}^{t} R_k^1(x^*) = \Delta_{2^n}, \tag{5.21}$$

the stabilization problem is solvable, denote the minimum number such that (5.21) holds as T^* and go to Step 5. Otherwise, set $t = t+1$. If $t > \omega$, go to Step 4. Otherwise, go to Step 3.
Step 4: For $t \geq \omega + 1$, construct the truth matrix $T^\omega_{R_{t-\omega}^1(x^*)} \in \mathfrak{B}_{2^m\times 2^n}$:

$$[T^\omega_{R_{t-\omega}^1(x^*)}]_{i,j} = \begin{cases} 1, \text{ if } L_\omega\delta_{2^m}^i\delta_{2^n}^j \in R_{t-\omega}^1(x^*), \\ 0, \text{ otherwise.} \end{cases} \tag{5.22}$$

Compute $R_{t-\omega+1}^\omega(x^*) = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T^\omega_{R_{t-\omega}^1(x^*)}) \neq \mathbf{0}_{2^m}\}$ and construct the truth matrix $T^{\omega-1}_{R_{t-\omega+1}^\omega(x^*)} \in \mathfrak{B}_{2^m\times 2^n}$:

$$[T^{\omega-1}_{R_{t-\omega+1}^\omega(x^*)}]_{i,j} = \begin{cases} 1, \text{ if } L_{\omega-1}\delta_{2^m}^i\delta_{2^n}^j \in R_{t-\omega+1}^\omega(x^*), \\ 0, \text{ otherwise.} \end{cases} \tag{5.23}$$

Compute $R_{t-\omega+2}^{\omega-1}(x^*) = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T^{\omega-1}_{R_{t-\omega+1}^\omega(x^*)}) \neq \mathbf{0}_{2^m}\}$. Repeat this procedure, constructing truth matrices $T^{\omega-2}_{R_{t-\omega+2}^{\omega-1}(x^*)}$, $T^{\omega-3}_{R_{t-\omega+3}^{\omega-2}(x^*)}, \cdots, T^{2}_{R_{t-2}^{3}(x^*)}$ respectively, and compute $R_{t-\omega+3}^{\omega-2}(x^*)$, $R_{t-\omega+4}^{\omega-3}(x^*)$, $\cdots$, $R_{t-1}^2(x^*)$.
Then, construct the truth matrix $T^1_{R_{t-1}^2(x^*)} \in \mathfrak{B}_{2^m\times 2^n}$:

$$[T^1_{R_{t-1}^2(x^*)}]_{i,j} = \begin{cases} 1, \text{ if } L_1\delta_{2^m}^i\delta_{2^n}^j \in R_{t-1}^2(x^*), \forall \delta_{2^n}^j \in \Delta_{2^n}\backslash(\bigcup_{k=1}^{t-1} R_k^1(x^*)), \\ 0, \text{ otherwise.} \end{cases} \tag{5.24}$$

Compute $R_t^1(x^*) = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T^1_{R^2_{t-1}(x^*)}) \neq \mathbf{0}_{2^m}\}$. If $R_t^1(x^*) \neq \emptyset$ and

$$\bigcup_{k=\omega+1}^{t} R_k^1(x^*) \bigcup \Lambda = \Delta_{2^n}, \tag{5.25}$$

where $\Lambda = \bigcup_{k=1}^{\omega} R_k^1(x^*) \neq \Delta_{2^n}$. The stabilization problem is solvable, denote the minimum number such that (5.25) holds as T^* and go to Step 5. Otherwise, set $t = t+1$. If $t > \omega \cdot (2^n - 1)$, the construction problem of stabilizer is not solvable, stop the algorithm. Otherwise, go to Step 4.

Step 5:

Case 1: If $T^* \leq \omega$, construct the switching-signal-dependent logical matrices $H_{\sigma(t)} \in \mathcal{L}_{2^m \times 2^n}$ as follows:

$$H_{l}|_{R_1^l(x^*)} \leq T^l_{R_0(x^*)|R_1^l(x^*)}, \ l \in [1 : T^*], \tag{5.26}$$

where

$$\mathrm{Col}_j(T^l_{R_0(x^*)|R_1^l(x^*)}) = \begin{cases} \mathrm{Col}_j(T^l_{R_0(x^*)}), \ \text{if } \delta_{2^n}^j \in R_1^l(x^*), \\ \mathbf{0}_{2^m}, \ \text{otherwise}, \end{cases}$$

and $l \in [1 : T^*]$. Moreover,

$$\begin{cases} H_{\vartheta}|_{R^{\vartheta}_{s_\vartheta}(x^*)} \leq T^{\vartheta}_{R^{\vartheta+1}_{s_\vartheta-1}(x^*)|R^{\vartheta}_{s_\vartheta}(x^*)}, \\ \vartheta \in [1 : \omega], \quad s_\vartheta \in [2 : T^* - (\vartheta - 1)], \end{cases} \tag{5.27}$$

where

$$\mathrm{Col}_j(T^{\vartheta}_{R^{\vartheta+1}_{s_\vartheta-1}(x^*)|R^{\vartheta}_{s_\vartheta}(x^*)}) = \begin{cases} \mathrm{Col}_j(T^{\vartheta}_{R^{\vartheta+1}_{s_\vartheta-1}(x^*)}), \ \text{if } \delta_{2^n}^j \in R^{\vartheta}_{s_\vartheta}(x^*), \\ \mathbf{0}_{2^m}, \ \text{otherwise}, \end{cases}$$

and $\vartheta \in [1 : \omega], \quad s_\vartheta \in [2 : T^* - (\vartheta - 1)]$. Besides,

$$H_{s}|_{R_0(x^*)} \leq T^s_{R_0(x^*)|R_0(x^*)}, \ s \in [T^* + 1 : \omega], \tag{5.28}$$

where $T^s_{R_0(x^*)|R_0(x^*)} = T^s_{R_0(x^*)}$ and $s \in [T^* + 1 : \omega]$.

Case 2: If $T^* \geq \omega + 1$, construct the switching signal-dependent logical matrices $H_{\sigma(t)} \in \mathcal{L}_{2^m \times 2^n}$ as follows:

$$H_{l}|_{R_1^l(x^*)} \leq T^l_{R_0(x^*)|R_1^l(x^*)}, \ l \in [1 : \omega], \tag{5.29}$$

where

$$\mathrm{Col}_j(T^l_{R_0(x^*)|R_1^l(x^*)}) = \begin{cases} \mathrm{Col}_j(T^l_{R_0(x^*)}), \ \text{if } \delta_{2^n}^j \in R_1^l(x^*), \\ \mathbf{0}_{2^m}, \ \text{otherwise}, \end{cases}$$

and $l \in [1:\omega]$. In addition,

$$\begin{cases} H_{\vartheta}|_{R^{\vartheta}_{s_\vartheta}(x^*)} \leq T^{\vartheta}_{R^{\vartheta+1}_{s_\vartheta-1}(x^*)|R^{\vartheta}_{s_\vartheta}(x^*)}, \\ \vartheta \in [1:\omega], \quad s_\vartheta = [2:T^* - (\vartheta - 1)], \end{cases} \tag{5.30}$$

where $R^{\omega+1}_k(x^*) = R^1_k(x^*)$, $k \geq 1$. Moreover,

$$\mathrm{Col}_j(T^{\vartheta}_{R^{\vartheta+1}_{s_\vartheta-1}(x^*)|R^{\vartheta}_{s_\vartheta}(x^*)}) = \begin{cases} \mathrm{Col}_j(T^{\vartheta}_{R^{\vartheta+1}_{s_\vartheta-1}(x^*)}), \text{ if } \delta^j_{2^n} \in R^{\vartheta}_{s_\vartheta}(x^*), \\ \mathbf{0}_{2^m}, \text{ otherwise}, \end{cases}$$

and $\vartheta \in [1:\omega], \quad s_\vartheta = [2:T^* - (\vartheta - 1)]$.

Remark 5.3.2. *If (5.21) or (5.25) holds, each initial state can reach x^*. Furthermore, from Algorithm 7, we have $\bigcap_{k=1}^{T^*} R^1_k(x^*) = \emptyset$, which ensures that T^* is the shortest time for all initial states to reach x^*.*

Figure 5.1 helps in explaining Algorithm 7.

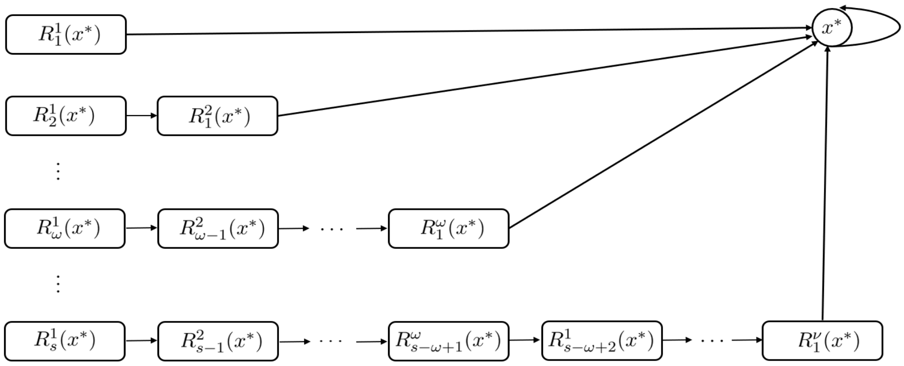

Figure 5.1: Illustration graph of Algorithm 7

Theorem 5.3.2. *Consider SBCNs (5.5) with periodic switching signal. The system is globally stabilizable to $x^* = \delta^\gamma_{2^n}$ if and only if Algorithm 7 reaches Step 5.*

Proof. (Sufficiency) Assume Algorithm 7 reaches Step 5, then we prove that SBCNs (5.5) are globally stabilizable to $x^* = \delta^\gamma_{2^n}$ under the switching-signal-dependent state feedback control $u_{\sigma(t)}(t) = H_{\sigma(t)}x(t)$, where the state feedback gain matrix $H_{\sigma(t)} \in \mathcal{L}_{2^m \times 2^n}$ is given by (5.26)-(5.28) or (5.29)-(5.30). From Step 1, suppose that $x^* \in \bigcap_{\nu=1}^{\omega} R^\nu_1(x^*)$, then x^* is a common control fixed point of the ω sub-systems. If $R^1_1(x^*) = \Delta_{2^n}$, then for any $x(0) \in \Delta_{2^n}$,

$u_1(t) = H_1 x(t)$ makes all initial states to reach x^* in one step and $u_\nu(t) = H_\nu x(t)$, $\nu \in [1:\omega]$ stabilizes the system at x^*.

Otherwise, if $R_1^1(x^*) \neq \Delta_{2^n}$ and $R_1^1(x^*) \bigcup R_2^1(x^*) = \Delta_{2^n}$, then for any $x(0) \in R_1^1(x^*)$, $u_1(t) = H_1 x(t)$ drives it to reach x^* in one step and $R_2^1(x^*)$ to reach $R_1^2(x^*)$ in one step. And $u_2(t) = H_2 x(t)$ drives $R_1^2(x^*)$ to reach x^* in one step. Moreover, $u_\nu(t) = H_\nu x(t)$, $\nu \in [1:\omega]$ stabilizes the system at x^*.

Continuing the above procedure, if there exists $s_0 \in [3:\omega]$, such that $\bigcup_{k=1}^{s_0-1} R_k^1(x^*) \neq \Delta_{2^n}$ and $\bigcup_{k=1}^{s_0} R_k^1(x^*) = \Delta_{2^n}$, then for any $x(0) \in R_1^1(x^*)$, $u_1(t) = H_1 x(t)$ drives it to reach x^* in one step and $R_2^1(x^*)$ to reach $R_1^2(x^*)$ in one step, similarly, it can drive $R_{s_0}^1(x^*)$ to reach $R_{s_0-1}^2(x^*)$ in one step. Moreover, $u_2(t) = H_2 x(t)$ drives $R_1^2(x^*)$ to reach x^* in one step; similarly, it can drive $R_{s_0-1}^2(x^*)$ to reach $R_{s_0-2}^3(x^*)$ in one step. Continuing the above discussion, $u_{s_0}(t) = H_{s_0} x(t)$ drives $R_1^{s_0}(x^*)$ to reach x^* in one step. Besides, $u_\nu(t) = H_\nu x(t)$, $\nu \in [1:\omega]$ stabilizes the system at x^*.

For the more general case, if $\bigcup_{k=1}^{\omega} R_k^1(x^*) \neq \Delta_{2^n}$ and there exists $s \geq \omega + 1$ such that $\bigcup_{k=1}^{s-1} R_k^1(x^*) \neq \Delta_{2^n}$ and $\bigcup_{k=1}^{s} R_k^1(x^*) = \Delta_{2^n}$, then for any $x(0) \in R_1^1(x^*)$, $u_1(t) = H_1 x(t)$ drives it to reach x^* in one step and $R_2^1(x^*)$ to reach $R_1^2(x^*)$ in one step. Similarly, it can drive $R_s^1(x^*)$ to reach $R_{s-1}^2(x^*)$ in one step. Moreover, $u_2(t) = H_2 x(t)$ drives $R_1^2(x^*)$ to reach x^* in one step and $R_{\omega-1}^2(x^*)$ to reach $R_{\omega-2}^3(x^*)$ in one step. Similarly, it can drive $R_{s-1}^2(x^*)$ to reach $R_{s-2}^3(x^*)$ in one step. Repeating the above steps, $u_\omega(t) = H_\omega x(t)$ drives $R_1^\omega(x^*)$ to reach x^* in one step, similarly, it can drive $R_{s-\omega+1}^\omega(x^*)$ to reach $R_{s-\omega+2}^1(x^*)$ in one step. Without loss of generality, there exists $u_\nu(t) = H_\nu x(t)$, such that $R_1^\nu(x^*)$ can reach x^* in one step. Furthermore, the system will be globally stabilized to x^* under the switching-signal-dependent state feedback controllers $u_\nu(t) = H_\nu x(t)$, $\nu \in [1:\omega]$.

(Necessity) Suppose now that system (5.5) is globally stabilizable to $x^* = \delta_{2^n}^\gamma$, and assume by contradiction that the Algorithm 7 does not reach to Step 5, that is $x^* \notin \bigcap_{\nu=1}^{\omega} R_1^\nu(x^*)$, *i.e.*, x^* is not a common control fixed point; or the equation (5.25) does not hold until $s = \omega \cdot (2^n - 1)$. It is obvious that the first case contradicts the condition that the system (5.5) is globally stabilizable to $x^* = \delta_{2^n}^\gamma$. For the second case, assume $R_{\omega\cdot(2^n-1)}^1(x^*) \neq \emptyset$ and

$$\bigcup_{k=1}^{\omega\cdot(2^n-1)} R_k^1(x^*) \neq \Delta_{2^n}.$$

That implies that there exists a state $\widehat{x} \in \Delta_{2^n} \backslash (\bigcup_{k=1}^{\omega\cdot(2^n-1)} R_k^1(x^*))$ such that no switching-signal-dependent state feedback control can drive it to $x^* = \delta_{2^n}^\gamma$, which also contradicts the condition that the system (5.5) is globally stabilizable to $x^* = \delta_{2^n}^\gamma$. □

Example 5.3.2. *With reference to system (5.13) in Example 5.3.1, we aim to design switching-signal-dependent state feedback controllers to stabilize system (5.13) to* $x^* = \delta_8^3$.

Based on the Algorithm 7, denote

$$R_0(x^*) = \{\delta_8^3\}.$$

Then the truth matrix $T^1_{R_0(x^*)} \in \mathfrak{B}_{4\times 8}$ can be constructed:

$$T^1_{R_0(x^*)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5.31}$$

From (5.31),

$$R_1^1(x^*) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^5, \delta_8^6, \delta_8^7\}.$$

Next, the truth matrix $T^2_{R_0(x^*)} \in \mathfrak{B}_{4\times 8}$ can be constructed

$$T^2_{R_0(x^*)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}. \tag{5.32}$$

From (5.32),

$$R_1^2(x^*) = \{\delta_8^1, \delta_8^3, \delta_8^5, \delta_8^7\}.$$

It is clear that $x^* \in R_1^1(x^*) \bigcap R_1^2(x^*)$. Then, construct truth matrix $T^1_{R_1^2(x^*)} \in \mathfrak{B}_{4\times 8}$ as

$$T^1_{R_1^2(x^*)} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5.33}$$

From (5.33),

$$R_2^1(x^*) = \{\delta_8^4, \delta_8^8\}.$$

Since $R_1^1(x^*) \bigcup R_2^1(x^*) = \Delta_8$, we have that $T^* = 2$, and we can construct the switching-signal-dependent state feedback gain matrices as

$$\begin{cases} H_1|_{R_1^1(x^*)} \leq T^1_{R_0(x^*)|R_1^1(x^*)}, \\ H_1|_{R_2^1(x^*)} \leq T^1_{R_1^2(x^*)|R_2^1(x^*)}, \end{cases} \tag{5.34}$$

where $T^1_{R_0(x^*)|R_1^1(x^*)} = T^1_{R_0(x^*)}$ and $T^1_{R_1^2(x^*)|R_2^1(x^*)} = T^1_{R_1^2(x^*)}$. Moreover,

$$H_2|_{R_1^2(x^*)} \leq T^2_{R_0(x^*)|R_1^2(x^*)}, \tag{5.35}$$

where $T^2_{R_0(x^*)|R_1^2(x^*)} = T^2_{R_0(x^*)}$.

From (5.34), it follows that all the possible choices of matrix $H_1 \in \mathcal{L}_{2^2\times 2^3}$ as follows

$$\begin{aligned} &\mathrm{Col}_j(H_1) \in \{\delta_4^3, \delta_4^4\},\ j \in [1:3], \\ &\mathrm{Col}_j(H_1) \in \{\delta_4^1, \delta_4^2\},\ j \in [5:8], \\ &\mathrm{Col}_4(H_1) \in \{\delta_4^1, \delta_4^2, \delta_4^3, \delta_4^4\}. \end{aligned} \tag{5.36}$$

Similarly, from (5.35), we have all the possible choices of matrix $H_2 \in \mathcal{L}_{2^2\times 2^3}$ as follows

$$\begin{aligned} &\mathrm{Col}_j(H_2) \in \{\delta_4^3, \delta_4^4\},\ j = 1, 3, \\ &\mathrm{Col}_j(H_2) \in \{\delta_4^1, \delta_4^2, \delta_4^3, \delta_4^4\},\ j = 5, 7. \end{aligned} \tag{5.37}$$

Putting (5.36) and (5.37) together, the switching-signal-dependent logical matrices $H_{\sigma(t)}$ are given by

$$H_{\sigma(t)} = \begin{cases} H_1, & t\ mod\ 2 = 0, \\ H_2, & t\ mod\ 2 = 1, \end{cases} \tag{5.38}$$

where $\mathrm{Col}_i(H_2)$, $i = 2, 4, 6, 8$ can be chosen arbitrarily.

5.4 Controller design for set stabilization of periodic SBCNs

In this section, we study whether the SBCNs with periodic switching signal can be set stabilized by switching-signal-dependent state feedback controller, and present the design algorithm via antecedence solution technique.

It must be noted that if the set Γ is reachable from some initial state, then there exists a control sequence such that it can be reachable within $\omega \cdot (2^n - r)$ steps, where $r = |\Gamma|$.

Similar to Algorithm 7, the switching-signal-dependent logical matrices $H_{\sigma(t)} \in \mathcal{L}_{2^m\times 2^n}$ which stabilize the system (5.5) to Γ can be constructed as follows.

Algorithm 8 Constructing switching-signal-dependent state feedback set stabilizers

Step 1: Let $R_0(\Gamma) = \Gamma$, and construct the truth matrices $T^{\nu}_{R_0(\Gamma)} \in \mathfrak{B}_{2^m\times 2^n}$, $\nu \in [1:\omega]$ as:

$$[T^{\nu}_{R_0(\Gamma)}]_{i,j} = \begin{cases} 1,\ \text{if}\ L_\nu \delta_{2^m}^i \delta_{2^n}^j \in \Gamma, \\ 0,\ \text{otherwise}. \end{cases} \tag{5.39}$$

Compute $R_1^{\nu}(\Gamma) = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T^{\nu}_{R_0(\Gamma)}) \neq \mathbf{0}_{2^m}\}$, $\nu \in [1:\omega]$ and check whether
$\Gamma \subseteq \bigcap_{\nu=1}^{\omega} R_1^{\nu}(\Gamma)$. If $\Gamma \nsubseteq \bigcap_{\nu=1}^{\omega} R_1^{\nu}(\Gamma)$, Γ is not a common control invariant set, and the

construction problem of stabilizer is not solvable, stop the algorithm. If $R_1^1(\Gamma) = \Delta_{2^n}$, set $T^* = 1$ and go to Step 5. Otherwise, go to Step 2.

Step 2: Construct the truth matrix $T^1_{R_1^2(\Gamma)} \in \mathfrak{B}_{2^m \times 2^n}$:

$$[T^1_{R_1^2(\Gamma)}]_{i,j} = \begin{cases} 1, \text{ if } L_1 \delta_{2^m}^i \delta_{2^n}^j \in R_1^2(\Gamma), \forall \delta_{2^n}^j \in \Delta_{2^n} \backslash R_1^1(\Gamma), \\ 0, \text{ otherwise.} \end{cases} \tag{5.40}$$

Compute $R_2^1(\Gamma) = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T^1_{R_1^2(\Gamma)}) \neq \mathbf{0}_{2^m}\}$. If $R_2^1(\Gamma) \neq \emptyset$ and $R_1^1(\Gamma) \bigcup R_2^1(\Gamma) = \Delta_{2^n}$, set $T^* = 2$ and go to Step 5. Otherwise, go to Step 3.

Step 3: For $t \in [3 : \omega]$, construct the truth matrix $T^{t-1}_{R_1^t(\Gamma)} \in \mathfrak{B}_{2^m \times 2^n}$:

$$[T^{t-1}_{R_1^t(\Gamma)}]_{i,j} = \begin{cases} 1, \text{ if } L_{t-1} \delta_{2^m}^i \delta_{2^n}^j \in R_1^t(\Gamma), \\ 0, \text{ otherwise.} \end{cases} \tag{5.41}$$

Compute $R_2^{t-1}(\Gamma) = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T^{t-1}_{R_1^t(\Gamma)}) \neq \mathbf{0}_{2^m}\}$ and construct the truth matrix $T^{t-2}_{R_2^{t-1}(\Gamma)} \in \mathfrak{B}_{2^m \times 2^n}$:

$$[T^{t-2}_{R_2^{t-1}(\Gamma)}]_{i,j} = \begin{cases} 1, \text{ if } L_{t-2} \delta_{2^m}^i \delta_{2^n}^j \in R_2^{t-1}(\Gamma), \\ 0, \text{ otherwise.} \end{cases} \tag{5.42}$$

Compute $R_3^{t-2}(\Gamma) = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T^{t-2}_{R_2^{t-1}(\Gamma)}) \neq \mathbf{0}_{2^m}\}$. Repeat this procedure, constructing truth matrices $T^{t-3}_{R_3^{t-2}(\Gamma)}$, $T^{t-4}_{R_4^{t-3}(\Gamma)}, \cdots$, $T^2_{R_{t-2}^3(\Gamma)}$ respectively, and compute $R_4^{t-3}(\Gamma)$, $R_5^{t-4}(\Gamma)$, $\cdots$, $R_{t-1}^2(\Gamma)$.

Then, construct the truth matrix $T^1_{R_{t-1}^2(\Gamma)} \in \mathfrak{B}_{2^m \times 2^n}$:

$$[T^1_{R_{t-1}^2(\Gamma)}]_{i,j} = \begin{cases} 1, \text{ if } L_1 \delta_{2^m}^i \delta_{2^n}^j \in R_{t-1}^2(\Gamma), \forall \delta_{2^n}^j \in \Delta_{2^n} \backslash (\bigcup_{k=1}^{t-1} R_k^1(\Gamma)), \\ 0, \text{ otherwise.} \end{cases} \tag{5.43}$$

Compute $R_t^1(\Gamma) = \{\delta_{2^n}^j \mid \mathrm{Col}_j(T^1_{R_{t-1}^2(\Gamma)}) \neq \mathbf{0}_{2^m}\}$. If $R_t^1(\Gamma) \neq \emptyset$ and

$$\bigcup_{k=1}^{t} R_k^1(\Gamma) = \Delta_{2^n}, \tag{5.44}$$

the set stabilization problem is solvable and denote the minimum number such that (5.44) holds as T^*, go to Step 5. Otherwise, set $t = t + 1$. If $t > \omega$, go to Step 4. Otherwise, go to Step 3.

Step 4: For $t \geq \omega + 1$, construct the truth matrix $T^{\omega}_{R_{t-\omega}^1(\Gamma)} \in \mathfrak{B}_{2^m \times 2^n}$:

$$[T^{\omega}_{R_{t-\omega}^1(\Gamma)}]_{i,j} = \begin{cases} 1, \text{ if } L_{\omega} \delta_{2^m}^i \delta_{2^n}^j \in R_{t-\omega}^1(\Gamma), \\ 0, \text{ otherwise.} \end{cases} \tag{5.45}$$

Compute $R^{\omega}_{t-\omega+1}(\Gamma) = \{\delta^j_{2^n} \mid \mathrm{Col}_j(T^{\omega}_{R^1_{t-\omega}(\Gamma)}) \neq \mathbf{0}_{2^m}\}$ and construct the truth matrix $T^{\omega-1}_{R^{\omega}_{t-\omega+1}(\Gamma)} \in \mathfrak{B}_{2^m\times 2^n}$:

$$[T^{\omega-1}_{R^{\omega}_{t-\omega+1}(\Gamma)}]_{i,j} = \begin{cases} 1, \text{ if } L_{\omega-1}\delta^i_{2^m}\delta^j_{2^n} \in R^{\omega}_{t-\omega+1}(\Gamma), \\ 0, \text{ otherwise.} \end{cases} \tag{5.46}$$

Compute $R^{\omega-1}_{t-\omega+2}(\Gamma) = \{\delta^j_{2^n} \mid \mathrm{Col}_j(T^{\omega-1}_{R^{\omega}_{t-\omega+1}(\Gamma)}) \neq \mathbf{0}_{2^m}\}$. Repeat this procedure, constructing truth matrices $T^{\omega-2}_{R^{\omega-1}_{t-\omega+2}(\Gamma)}$, $T^{\omega-3}_{R^{\omega-2}_{t-\omega+3}(\Gamma)}, \cdots, T^2_{R^3_{t-2}(\Gamma)}$ respectively, and compute $R^{\omega-2}_{t-\omega+3}(\Gamma)$, $R^{\omega-3}_{t-\omega+4}(\Gamma)$, $\cdots$, $R^2_{t-1}(\Gamma)$.
Then, construct the truth matrix $T^1_{R^2_{t-1}(\Gamma)} \in \mathfrak{B}_{2^m\times 2^n}$:

$$[T^1_{R^2_{t-1}(\Gamma)}]_{i,j} = \begin{cases} 1, \text{ if } L_1\delta^i_{2^m}\delta^j_{2^n} \in R^2_{t-1}(\Gamma), \forall \delta^j_{2^n} \in \Delta_{2^n}\backslash(\bigcup_{k=1}^{t-1} R^1_k(\Gamma)), \\ 0, \text{ otherwise.} \end{cases} \tag{5.47}$$

Compute $R^1_t(\Gamma) = \{\delta^j_{2^n} \mid \mathrm{Col}_j(T^1_{R^2_{t-1}(\Gamma)}) \neq \mathbf{0}_{2^m}\}$. If $R^1_t(\Gamma) \neq \emptyset$ and

$$\bigcup_{k=\omega+1}^{t} R^1_k(\Gamma) \bigcup \Lambda = \Delta_{2^n}, \tag{5.48}$$

where $\Lambda = \bigcup_{k=1}^{\omega} R^1_k(\Gamma) \neq \Delta_{2^n}$. In this case, the set stabilization problem is solvable, denote the minimum number such that (5.48) holds as T^*, and go to step 5. Otherwise, set $t = t+1$. If $t > \omega \cdot (2^n - r)$, the construction problem of set stabilizer is not solvable, stop the algorithm. Otherwise, go to Step 4.

Step 5:

Case 1: If $T^* \leq \omega$, construct the switching-signal-dependent logical matrices $H_{\sigma(t)} \in \mathcal{L}_{2^m\times 2^n}$ as follows:

$$H_{l}|_{R^l_1(\Gamma)} \leq T^l_{R_0(\Gamma)|R^l_1(\Gamma)}, \ l \in [1 : T^*], \tag{5.49}$$

where

$$\mathrm{Col}_j(T^l_{R_0(\Gamma)|R^l_1(\Gamma)}) = \begin{cases} \mathrm{Col}_j(T^l_{R_0(\Gamma)}), \text{ if } \delta^j_{2^n} \in R^l_1(\Gamma), \\ \mathbf{0}_{2^m}, \text{ otherwise,} \end{cases}$$

and $l \in [1 : T^*]$. Moreover,

$$\begin{cases} H_{\vartheta}|_{R^{\vartheta}_{s_\vartheta}(\Gamma)} \leq T^{\vartheta}_{R^{\vartheta+1}_{s_\vartheta-1}(\Gamma)|R^{\vartheta}_{s_\vartheta}(\Gamma)}, \\ \vartheta \in [1 : \omega], \quad s_\vartheta \in [2 : T^* - (\vartheta - 1)], \end{cases} \tag{5.50}$$

where

$$\mathrm{Col}_j(T^{\vartheta}_{R^{\vartheta+1}_{s_\vartheta-1}(\Gamma)|R^{\vartheta}_{s_\vartheta}(\Gamma)}) = \begin{cases} \mathrm{Col}_j(T^{\vartheta}_{R^{\vartheta+1}_{s_\vartheta-1}(\Gamma)}), \text{ if } \delta^j_{2^n} \in R^{\vartheta}_{s_\vartheta}(\Gamma), \\ \mathbf{0}_{2^m}, \text{ otherwise}, \end{cases}$$

and $\vartheta \in [1:\omega], \quad s_\vartheta \in [2 : T^* - (\vartheta - 1)]$. Besides,

$$H_{s|_{R_0(\Gamma)}} \leq T^s_{R_0(\Gamma)|R_0(\Gamma)}, \; s \in [T^* + 1 : \omega], \tag{5.51}$$

where $T^s_{R_0(\Gamma)|R_0(\Gamma)} = T^s_{R_0(\Gamma)}$, $s \in [T^* + 1 : \omega]$.
Case 2: If $T^* \geq \omega + 1$, construct the switching-signal-dependent logical matrices $H_{\sigma(t)} \in \mathcal{L}_{2^m \times 2^n}$ as follows:

$$H_{l|_{R^l_1(\Gamma)}} \leq T^l_{R_0(\Gamma)|R^l_1(\Gamma)}, \; l \in [1 : \omega], \tag{5.52}$$

where

$$\mathrm{Col}_j(T^l_{R_0(\Gamma)|R^l_1(\Gamma)}) = \begin{cases} \mathrm{Col}_j(T^l_{R_0(\Gamma)}), \text{ if } \delta^j_{2^n} \in R^l_1(\Gamma), \\ \mathbf{0}_{2^m}, \text{ otherwise}, \end{cases}$$

and $l \in [1 : \omega]$. In addition,

$$\begin{cases} H_{\vartheta|_{R^{\vartheta}_{s_\vartheta}(\Gamma)}} \leq T^{\vartheta}_{R^{\vartheta+1}_{s_\vartheta-1}(\Gamma)|R^{\vartheta}_{s_\vartheta}(\Gamma)}, \\ \vartheta \in [1 : \omega], \quad s_\vartheta \in [2 : T^* - (\vartheta - 1)], \end{cases} \tag{5.53}$$

where $R^{\omega+1}_k(\Gamma) = R^1_k(\Gamma)$, $k \geq 1$. Moreover,

$$\mathrm{Col}_j(T^{\vartheta}_{R^{\vartheta+1}_{s_\vartheta-1}(\Gamma)|R^{\vartheta}_{s_\vartheta}(\Gamma)}) = \begin{cases} \mathrm{Col}_j(T^{\vartheta}_{R^{\vartheta+1}_{s_\vartheta-1}(\Gamma)}), \text{ if } \delta^j_{2^n} \in R^{\vartheta}_{s_\vartheta}(\Gamma), \\ \mathbf{0}_{2^m}, \text{ otherwise}, \end{cases}$$

and $\vartheta \in [1:\omega], \quad s_\vartheta \in [2 : T^* - (\vartheta - 1)]$.

Similarly, here T^* is the shortest time for all initial states to reach Γ.

Corollary 5.4.1. *Consider SBCNs (5.5) with periodic switching signal. Then the system is globally stabilizable to a set Γ if and only if the Algorithm 8 reaches Step 5.*

Example 5.4.1. *Consider system (5.10) in Example 5.3.1. Let $U_1(t) \equiv 0$ and we obtain system (5.11). Fix $U_2(t) \equiv 1$ and let $V_1(t) = U_1(t)$, $V_2(t) = U_3(t)$, then system (5.10) can be converted to*

$$\begin{cases} X_1(t+1) = \neg V_1(t) \wedge (X_2(t) \vee X_3(t)), \\ X_2(t+1) = \neg V_1(t) \wedge X_1(t), \\ X_3(t+1) = \neg V_1(t). \end{cases} \tag{5.54}$$

Assume that system (5.11) and (5.54) are sub-systems of a SBCN with periodic switching law (5.14) and the algebraic formulation of SBCN can be expressed as

$$x(t+1) = L\sigma(t)v(t)x(t),$$

where $L = [L_1\ L_2]$, and

$$\begin{aligned} L_1 &= \delta_8[1\ 1\ 1\ 5\ 3\ 3\ 3\ 7\ 1\ 1\ 1\ 5\ 3\ 3\ 3\ 7\ 3\ 3\ 3\ 7\ 4\ 4\ 4\ 8\ 3\ 3\ 3\ 7\ 4\ 4\ 4\ 8], \\ L_2 &= \delta_8[8\ 8\ 8\ 8\ 8\ 8\ 8\ 8\ 8\ 8\ 8\ 8\ 8\ 8\ 8\ 8\ 1\ 1\ 1\ 5\ 3\ 3\ 3\ 7\ 1\ 1\ 1\ 5\ 3\ 3\ 3\ 7]. \end{aligned}$$

It is obvious that $x^* = \delta_8^3$ is not a common control fixed point. Then, we aim to design switching-signal-dependent state feedback controllers to stabilize the new SBCN to a set Γ which contains $x^* = \delta_8^3$. In the following, we consider $\Gamma = \{\delta_8^1, \delta_8^3, \delta_8^8\}$.

Based on the Algorithm 8, we have

$$R_0(\Gamma) = \Gamma = \{\delta_8^1, \delta_8^3, \delta_8^8\}.$$

Then the matrix $T^1_{R_0(\Gamma)}$ can be constructed

$$T^1_{R_0(\Gamma)} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.55}$$

From (5.55),

$$R_1^1(\Gamma) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^5, \delta_8^6, \delta_8^7, \delta_8^8\}.$$

Next, the matrix $T^2_{R_0(\Gamma)}$ can be constructed as

$$T^2_{R_0(\Gamma)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{5.56}$$

From (5.56),

$$R_1^2(\Gamma) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^4, \delta_8^5, \delta_8^6, \delta_8^7, \delta_8^8\}.$$

It is clear that

$$\Gamma \subseteq R_1^1(\Gamma) \bigcap R_1^2(\Gamma).$$

Then, construct $T^1_{R_1^2(\Gamma)}$ as

$$T^1_{R_1^2(\Gamma)} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5.57}$$

From (5.57),

$$R_2^1(\Gamma) = \{\delta_8^4\}.$$

Since $R_1^1(\Gamma)\bigcup R_2^1(\Gamma) = \Delta_8$, then $T^* = 2$, and we can construct the switching-signal-dependent state feedback gain matrices as

$$\begin{cases} H_1|_{R_1^1(\Gamma)} \leq T^1_{R_0(\Gamma)|R_1^1(\Gamma)}, \\ H_1|_{R_2^1(\Gamma)} \leq T^1_{R_1^2(\Gamma)|R_2^1(\Gamma)}, \end{cases} \tag{5.58}$$

where $T^1_{R_0(\Gamma)|R_1^1(\Gamma)} = T^1_{R_0(\Gamma)}$ and $T^1_{R_1^2(\Gamma)|R_2^1(\Gamma)} = T^1_{R_1^2(\Gamma)}$. Moreover,

$$H_2|_{R_1^2(\Gamma)} \leq T^2_{R_0(\Gamma)|R_1^2(\Gamma)}, \tag{5.59}$$

where $T^2_{R_0(\Gamma)|R_1^2(\Gamma)} = T^2_{R_0(\Gamma)}$.

From (5.58), it follows that all the possible choices of matrix $H_1 \in \mathcal{L}_{2^2\times 2^3}$ as follows:

$$\begin{aligned} &\text{Col}_j(H_1) \in \{\delta_4^1, \delta_4^2, \delta_4^3, \delta_4^4\},\ j \in [1:4], \\ &\text{Col}_j(H_1) \in \{\delta_4^1, \delta_4^2\},\ j \in [5:7], \\ &\text{Col}_8(H_1) = \delta_4^4 \in \{\delta_4^3, \delta_4^4\}. \end{aligned} \tag{5.60}$$

Similarly, from (5.59), we have all the possible choices of matrix $H_2 \in \mathcal{L}_{2^2\times 2^3}$ as follows:

$$\begin{aligned} &\text{Col}_j(H_2) = \delta_4^1 \in \{\delta_4^1, \delta_4^2, \delta_4^3, \delta_4^4\}, \\ &j = 1, 2, 3, 5, 6, 7, \\ &\text{Col}_j(H_2) = \delta_4^3 \in \{\delta_4^1, \delta_4^2\},\ j = 4, 8. \end{aligned} \tag{5.61}$$

Putting (5.60) and (5.61) together, the switching-signal-dependent logical matrices $H_{\sigma(t)}$ are given by

$$H_{\sigma(t)} = \begin{cases} H_1, & t \ mod\ 2 = 0, \\ H_2, & t \ mod\ 2 = 1, \end{cases} \tag{5.62}$$

which can stabilize the SBCN to $\Gamma = \{\delta_8^1, \delta_8^3, \delta_8^8\}$.

5.5 Conclusions

In this chapter, we have investigated the stabilization and set stabilization problems of SBCNs under periodic switching signal. Necessary and sufficient conditions to detect the solvability of stabilization and set stabilization problems of SBCNs under periodic switching signal have been derived. Moreover, a constructive procedure has been presented to design open loop controller. Using an antecedence solution technique, the design algorithms have been established to construct switching-signal-dependent state feedback controllers. Illustrative examples have been presented to show the effectiveness of the obtained results.

Chapter 6

Event-triggered Control Design for Networked Evolutionary Games with Time-Invariant Delay in Strategies

6.1 Introduction

It is well known that the dynamics of NEGs may converge to several different equilibria [155], which may lead to undesired benefits for players. Thus, it is necessary to develop control means in the investigation of NEGs. In the early stage of the study for controlled NEGs, pseudo-players were always regarded as control to influence the evolutionary dynamics of the game, whose strategies can be assigned freely [18, 32, 140, 29, 139]. However, in these literature, the control inputs need to be executed at each time instant. It is no doubt that this control paradigm is a waste of resources. In reaction to this problem, event-triggered controls have been considered for the study of NEGs in [34]. Differently from normal controls, in [34], the control only works for some certain strategy profiles, which can be regarded as event-triggered conditions. The main advantage of event-triggered control is that the costs for control can be reduced and the evolutionary dynamics can be remain at the same time. Up to now, this kind of control has been applied to the investigation of set stabilization [81], disturbance decoupling problem of BNs and MVLNs [55, 120].

It must be noted that the NEGs mentioned in the above are considered without time delays in strategies. However, the time delays phenomenon is very common in real world, and logical networks with time delays have been studied [85, 146]. It is recognized that the interactions between players can not take place instantaneously and their reactions can not be immediate, which will unavoidable cause time delays in strategies. Furthermore, time delays may result in some undesired performance of the games and make the analysis of evolutionary dynamics much more complicated. Thus, the investigation of DNEGs is a very significant topic. Wang et al have investigated the modeling and stability of a class of finite evolutionary games with time delays in strategies [121], and a sufficient condition to assure the stability of the delayed evolutionary potential games at a pure Nash equilibrium was derived. To the best of our knowledge, the DNEGs have not been fully investigated.

This chapter first studies the dynamics and control problems of DNEGs, and an event-triggered state feedback controller is designed to guarantee the global convergence of the desired strategy profile. Note that the DNEGs can be regarded as a delayed logical dynamic system. By using the STP method, the delayed logical dynamic system can be converted into a conventional delayed discrete time system with algebraic form, which makes possible to use the classical control theory and method to analyse evolutionary dynamics and design controller for DNEGs. The main contributions of this chapter are:

- The dynamics of DNEGs are converted into algebraic forms.
- An event-triggered state feedback controller is constructed, and necessary and sufficient conditions for the global convergence of the desired strategy profile of the DNEGs are derived.

6.2 Dynamics analysis of DNEGs

Assume that all strategies have a time-invariant delay $\tau + 1$ in the NEGs, that is, the strategy of player i at time $t+1$ depends on the behaviors of all players at time $t-\tau$. This can be described as

$$x_i(t+1) = f_i(x_1(t-\tau), x_2(t-\tau), \cdots, x_n(t-\tau)), \quad i \in N, \tag{6.1}$$

where $\tau \in \mathbb{Z}_+$ and f_i is determined by the strategy adjustment rule.

When the topology structure of the network is considered, then, for any player $i \in N$, his strategy updating rule based on local information can be expressed with a set of mappings:

$$x_i(t+1) = f_i\Big(\{x_j(t-\tau) | j \in U(i)\}\Big),\ t \geq 0,\ i \in N, \tag{6.2}$$

where $x_j(t-\tau)$ denotes the strategy of player j at time $t-\tau$, and $U(i)$ is the set of neighborhood nodes of player i.

The payoff of each player can be calculated as follows:

$$p_i(t) = \sum_{j\in U(i)} p_{ij}(x_i(t), x_j(t)),\ i \in N,$$

where $p_{ij}(x_i(t), x_j(t)) : S_0 \times S_0 \longrightarrow \mathbb{R}$ is the payoff of player i playing with his neighbor j at time t when i takes strategy $x_i(t)$ and j takes strategy $x_j(t)$, and S_0 is the strategy set.

Note that when the network graph and FNG are selected, the strategy profile dynamics are uniquely determined by the strategy updating rule. In this chapter we consider an MBRAR with minimum priority [23], which is described as:

$$BR_i := \arg\max_{x_i\in S_0} p_i(x_i, x_j(t) \mid j \in U(i)). \tag{6.3}$$

Here, each player forecasts that his opponents will repeat their present strategies, and the strategy choice at next time is the best response against his neighbors' strategies of the present step. Moreover, if $x_i(t) \in BR_i$, then $x_i(t + 1) = x_i(t)$. If the strategies with best payoff are not unique, that is, $|BR_i| > 1$, then player i chooses one corresponding to the minimum priority: $x_i(t + 1) = \min\{x|x \in BR_i\}$.

For DNEGs, we use a Parallel MBRAR as strategy updating rule, that is, all the players update their strategies simultaneously. Based on Lemma 1.2.1, we can obtain the algebraic form of the evolutionary dynamics of each player as follows

$$x_i(t + 1) = M_i x(t - \tau),\ i \in [1 : n], \tag{6.4}$$

where $M_i \in \mathcal{L}_{k\times k^n}$ and $x(t - \tau) = \ltimes_{i=1}^{n} x_i(t - \tau)$.

Based on (6.4), the algebraic formulation of dynamics for the DNEGs can be expressed as

$$x(t + 1) = Lx(t - \tau), \tag{6.5}$$

where $L = M_1 * M_2 * \cdots * M_n \in \mathcal{L}_{k^n\times k^n}$.

Note that there is a standard procedure [32] to convert the evolutionary dynamics of each player into its algebraic form (6.4). In fact, (6.5) is exactly the algebraic form of a DKVLN. In other words, the dynamics of the DNEGs are equivalently expressed into a DKVLN. In the following, the results of stability analysis and control of DKVLNs in Chapter 2 can be applied to study DNEGs easily.

Remark 6.2.1. *The evolutionary dynamics of system (6.5) are affected by the $\tau+1$ initial strategy profiles $x(-\tau), x(-\tau+1), \cdots, x(0)$. That is the main difference with NEGs without delay.*

6.3 Control design of DNEGs

In this section, we consider necessary and sufficient conditions for the evolutionary dynamics to globally converge to the only desired strategy profile. Moreover, the strategies of the control players are presented.

In order to reduce the control task execution times and costs, and obtain the desired evolutionary performance of the game, an event-triggered control is considered for DNEGs. This kind of control can be regarded as an intermittent control, which only works for some specific strategy profiles.

Without loss of generality, we assume that $U = \{1, 2, \cdots, m\}$ is the set of players who are affected by external intermittent controls and Γ is the set of special events in which the control can be triggered.

Based on the above analysis, if $x(t-\tau) \in \Gamma$, that is, the event-triggered condition occurs, then the algebraic form of DNEGs under control can be formulated as

$$x(t+1) = \overline{L}u(t-\tau)x(t-\tau), \tag{6.6}$$

for some $\overline{L} \in \mathcal{L}_{k^n \times k^{n+m}}$, $u(t-\tau) = \ltimes_{j=1}^{m} u_j(t-\tau)$ and $x(t-\tau) = \ltimes_{i=1}^{n} x_i(t-\tau)$.

If $x(t-\tau) \notin \Gamma$, there is no control action and the game dynamics will evolve naturally. In this situation the corresponding algebraic form is (6.5).

To obtain the general algebraic form of DNEGs under event-triggered control, we can regard the no control action as a special control strategy u_0, and let $u_0 \sim \delta_{k^m+1}^{k^m+1}$, the control strategy $\delta_{k^m}^{i} \sim \delta_{k^m+1}^{i}$, $i \in [1 : k^m]$. Thus, the algebraic form of DNEGs under event-triggered control can be formulated as

$$x(t+1) = [\overline{L} \;\; L]\widetilde{u}(t-\tau)x(t-\tau), \tag{6.7}$$

where $\widetilde{u}(t-\tau) = u_0(t-\tau)$, if $x(t-\tau) \in \Delta_{k^n} \setminus \Gamma$. Otherwise, $\widetilde{u}(t-\tau) = u(t-\tau)$.

Assume $x^* = \delta_{k^n}^{\theta}$ be the desired only strategy profile of DNEGs, such that

$$Lx^* = x^*. \tag{6.8}$$

The objective is to design an event-triggered state feedback controller in the following algebraic form

$$\widetilde{u}(t) = Hx(t), \tag{6.9}$$

where $H \in \mathcal{L}_{(k^m+1)\times k^n}$, such that under the controller (6.9), the dynamics of DNEGs (6.7) are global convergent to $x^* = \delta_{k^n}^{\theta}$.

It is obviously that the designing problem of state feedback event-triggered controller becomes the construction of feedback control matrix $H \in \mathcal{L}_{(k^m+1)\times k^n}$. Plugging (6.9) into (6.7), we can obtain that

$$\begin{aligned} x(t) &= [\overline{L} \;\; L]\widetilde{u}(t-\tau-1)x(t-\tau-1) \\ &= [\overline{L} \;\; L]Hx(t-\tau-1)x(t-\tau-1) \\ &= [\overline{L} \;\; L]H\Psi_{k^n}x(t-\tau-1), \end{aligned} \tag{6.10}$$

where $[\overline{L} \;\; L] \in \mathcal{L}_{k^n \times (k^m+1)k^n}$ and $x(t-\tau-1) = \ltimes_{i=1}^{n} x_i(t-\tau-1) \in \Delta_{k^n}$.

Let $y(t) = \ltimes_{l=t+\tau}^{t} x(l) \in \Delta_{k^{(\tau+1)n}}$. Then (6.10) is transformed into

$$\begin{aligned} y(t+1) &= x(t+\tau+1)x(t+\tau)\cdots x(t+1) \\ &= [\overline{L} \;\; L]H\Psi_{k^n}x(t)x(t+\tau)\cdots x(t+1) \\ &= [\overline{L} \;\; L]H\Psi_{k^n}W_{[k^{\tau n},k^n]}y(t) \\ &:= \Phi y(t), \end{aligned} \tag{6.11}$$

where $\Phi = [\overline{L} \;\; L]H\Psi_{k^n}W_{[k^{\tau n},k^n]} \in \mathcal{L}_{k^{(\tau+1)n}\times k^{(\tau+1)n}}$. Thus, we have

$$y(t) = \Phi^{t+\tau}y(-\tau). \tag{6.12}$$

The following theorem states necessary and sufficient conditions to determine whether the evolutionary dynamics of (6.7) can globally converge to the desired only strategy profile under the event-triggered control.

Theorem 6.3.1. *Consider DNEGs (6.7) under the state feedback event-triggered controller (6.9) with initial strategy profiles $x(-\tau), x(-\tau+1), \cdots, x(0) \in \Delta_{k^n}$. The evolutionary dynamics of (6.7) converge to the strategy profile $x^* = \delta_{k^n}^{\theta}$ globally, if and only if there exist a logical matrix $H \in \mathcal{L}_{(k^m+1)\times k^n}$ and an integer $T \in [\tau+1 : (\tau+1)(k^n-1)]$ such that*

$$\mathrm{Col}(\Phi^T) = \{(\delta_{k^n}^{\theta})^{\tau+1}\}, \tag{6.13}$$

where $\Phi = [\overline{L} \;\; L]HM_{k^n}^{r}W_{[k^{\tau n},k^n]} \in \mathcal{L}_{k^{(\tau+1)n}\times k^{(\tau+1)n}}$ and $(\delta_{k^n}^{\theta})^{\tau+1} = \underbrace{\delta_{k^n}^{\theta} \ltimes \delta_{k^n}^{\theta} \ltimes \cdots \ltimes \delta_{k^n}^{\theta}}_{\tau+1}$.

Proof. (Necessity) Assume that the evolutionary dynamics of (6.7) globally converge to the strategy profile $x^* = \delta_{k^n}^{\theta}$ under the controller (6.9). Then there exists an integer $M \in [0 : (\tau+1)(k^n-2)+1]$ such that

$$x(t) = \delta_{k^n}^{\theta}, \forall t \geq M.$$

Thus, we have

$$y(M) = x(M+\tau)x(M+\tau-1)\cdots x(M) = (\delta_{k^n}^{\theta})^{\tau+1}.$$

Since $y(M) = \Phi^{M+\tau}y(-\tau)$ holds for any initial strategy profiles $y(-\tau) = \ltimes_{i=0}^{-\tau}x(i)$, we have

$$\mathrm{Col}(\Phi^{M+\tau}) = \{(\delta_{k^n}^{\theta})^{\tau+1}\}.$$

Choosing $T = M + \tau$, (6.13) is satisfied.

(Sufficiency) Suppose that (6.13) is satisfied. We will prove that the evolutionary dynamics of (6.7) globally converge to the strategy profile $x^* = \delta_{k^n}^{\theta}$ under controller $\widetilde{u}(t) = Hx(t)$. Since $y(t) = \ltimes_{i=t+\tau}^{t}x(i) \in \Delta_{k^{(\tau+1)n}}$, then

$$y(T-\tau) = \Phi^{T}y(-\tau) = (\delta_{k^n}^{\theta})^{\tau+1}.$$

For any initial strategy profiles $x(-\tau), x(-\tau+1), \cdots, x(0) \in \Delta_{k^n}$, and $\forall t \geq T-\tau$, we have

$$\begin{aligned}
x(t) &= D_r^{k^{\tau n},k^n}x(t+\tau)x(t+\tau-1)\cdots x(t)\\
&= D_r^{k^{\tau n},k^n}y(t)\\
&= D_r^{k^{\tau n},k^n}\Phi^{t+\tau}y(-\tau)\\
&= D_r^{k^{\tau n},k^n}\Phi^{T}\Phi^{t+\tau-T}y(-\tau)\\
&= D_r^{k^{\tau n},k^n}\Phi^{T}[\Phi^{t+\tau-T}\ltimes_{i=0}^{-\tau}x(i)]\\
&= D_r^{k^{\tau n},k^n}(\delta_{k^n}^{\theta})^{\tau+1}\\
&= \delta_{k^n}^{\theta}.
\end{aligned}$$

Thus, the evolutionary dynamics will converge to the strategy profile $x^* = \delta_{k^n}^{\theta}$ globally under controller $\widetilde{u}(t) = Hx(t)$. □

Similarly, we have the following corollary.

Corollary 6.3.2. *The evolutionary dynamics of (6.7) globally converge to the desired only final strategy profile $x^* = \delta_{k^n}^{\theta}$ under controller (6.9) for any initial strategy profiles $x(-\tau), x(-\tau+1), \cdots, x(0) \in \Delta_{k^n}$, if and only if there exist a logical matrix $H \in \mathcal{L}_{(k^m+1)\times k^n}$ and an integer $\alpha \in [1 : k^n - 1]$ such that*

$$\mathrm{Col}(([\overline{L}\ \ L]HM_{k^n}^{r})^{\alpha}) = \{\delta_{k^n}^{\theta}\}. \tag{6.14}$$

From Theorem 6.3.1 or Corollary 6.3.2, the controller design problem for event-triggered state feedback controller is equivalent to solve (6.13) or (6.14). Since the solution $H \in \mathcal{L}_{(k^m+1)\times k^n}$ is very difficult to calculate directly, we next try to design the state feedback event-triggered controller by calculating the reachable sets.

First, from (6.6), we can split the matrix $\widehat{L} = \overline{L}W_{[k^n,k^m]}$ into k^n blocks

$$\widehat{L} = [\mathrm{Blk}_1(\widehat{L})\ \mathrm{Blk}_2(\widehat{L})\ \cdots\ \mathrm{Blk}_{k^n}(\widehat{L})],$$

where $\mathrm{Blk}_i(\widehat{L}) \in \mathcal{L}_{k^n\times k^m}$, $i \in [1 : k^n]$.

Then, we introduce two kinds of reachable sets as follows:

$$E_r(\delta_{k^n}^{i_0}) = \{\delta_{k^n}^{j} \mid \exists r \in [1 : k^n], \text{ such that } \mathrm{Col}_j(L^r) = \delta_{k^n}^{i_0}\},$$

$$\widehat{E}(\delta_{k^n}^{i_0}) = \{\delta_{k^n}^{j} \mid \delta_{k^n}^{i_0} \in \mathrm{Col}(\mathrm{Blk}_j(\widehat{L})), j \in [1 : k^n]\}.$$

It is easy to recognize that $E_r(\delta_{k^n}^{i_0})$ is the set of all the strategy profiles which can evolve to $\delta_{k^n}^{i_0}$ naturally in r steps, while $\widehat{E}(\delta_{k^n}^{i_0})$ denotes the set of all the strategy profiles which can be steered to $\delta_{k^n}^{i_0}$ by control in one step.

In particular, if $S = \{\delta_{k^n}^{i_1}, \delta_{k^n}^{i_2}, \cdots, \delta_{k^n}^{i_l}\}$, then we define

$$E_r(S) = \bigcup_{\mu=1}^{l} E_r(\delta_{k^n}^{i_\mu}),$$

$$\widehat{E}(S) = \bigcup_{\mu=1}^{l} \widehat{E}(\delta_{k^n}^{i_\mu}).$$

Next, for the desired strategy profile $x^* = \delta_{k^n}^{\theta}$, a sequence of set of vectors is defined as follows.

$$(i) \quad S_0 = E_r(\delta_{k^n}^{\theta}),$$

$$(ii) \quad S_i = \widehat{S}_i \cup \widetilde{S}_i, i \geq 1, \tag{6.15}$$

where $\widehat{S}_i = \widehat{E}(S_{i-1}) \bigcap [\Delta_{k^n} \setminus \bigcup_{\mu=0}^{i-1} S_\mu]$, $\widetilde{S}_i = E_r(\widehat{S}_i) \setminus \widehat{S}_i$.

We can easily see from (6.15), all the strategy profiles in S_0 can naturally evolve to $x^* = \delta_{k^n}^{\theta}$, in which the control scheme will not be triggered. When the given profile $x(t-\tau) \notin S_0$, it is complicated to detect whether the control scheme will be triggered or not. In fact, the event-triggered scheme can be represented as Figure 6.1.

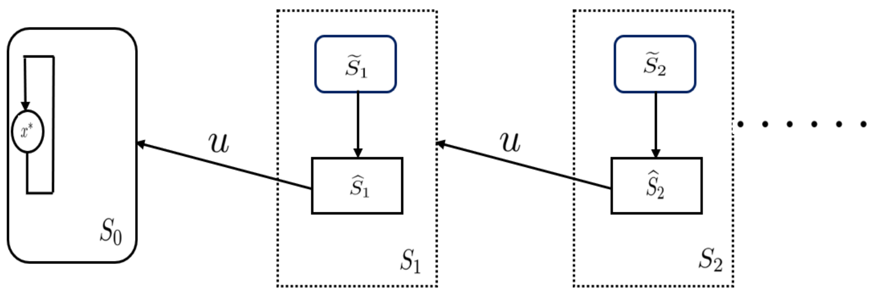

Figure 6.1: The illustration graph of event-triggered scheme

First, we can determine the profiles which will be steered to S_0 by control in one step, and denote the set of these profiles as $\widehat{S}_1$, in which all the elements will trigger the control scheme. When $x(t-\tau) \notin S_0 \bigcup \widehat{S}_1$, then we need to detect whether $x(t-\tau)$ can naturally converge to $\widehat{S}_1$ or not. If $x(t-\tau)$ can converge to $\widehat{S}_1$ naturally, then $x(t-\tau)$ will not trigger the control scheme to reach $\widehat{S}_1$, and we denote the set of all these states as $\widetilde{S}_1$. After the states in $\widetilde{S}_1$ evolving to $\widehat{S}_1$, the states will be steered to S_0 by control in one step. Moreover, let $S_1 = \widehat{S}_1 \bigcup \widetilde{S}_1$. If $x(t-\tau)$ can not converge to $\widehat{S}_1$ naturally, then we need to find the profiles which will be steered to S_1 by control in one step, and denote the set of these profiles as $\widehat{S}_2$, in which all the elements will trigger the control scheme. Similarly, we can continue the above procedure to obtain the triggered conditions.

From the above procedure we have that, $\forall i, j \in \{0, 1, 2, \cdots\}$, $i \neq j$, $S_i \cap S_j = \emptyset$. If $S_{j_0} = \emptyset$, for some $j_0 \geq 1$, then $S_j = \emptyset, \forall j \geq j_0$. Since there exist only finite strategy profiles, then there exists a positive number $T < k^n$, such that $S_{T+1} = \emptyset$ and $\sum_{i=0}^{T} |S_i| \leq k^n$, the constructing procedure for $S_i, i \geq 0$, will terminate in most k^n steps.

Based on the above discussion, we have the following result.

Theorem 6.3.3. *The evolutionary dynamics of (6.7) globally converge to the strategy profile $x^* = \delta_{k^n}^{\theta}$ under a state feedback event-triggered controller (6.9), if and only if there exists an integer $T \in [1 : k^n - 1]$ such that $\sum_{i=0}^{T} |S_i| = k^n$.*

Proof. (Necessity) Assume that the evolutionary dynamics of (6.7) globally converge to the strategy profile $x^* = \delta_{k^n}^{\theta}$ under a state feedback event-triggered controller (6.9). Then all the strategy profiles converge to $x^* = \delta_{k^n}^{\theta}$, and there exists an integer T such that $\sum_{i=0}^{T} |S_i| = k^n$. Let T be the smallest positive number such that $\sum_{i=0}^{T} |S_i| = k^n$, then we will prove that $T \leq k^n - 1$.

It is enough to show that $\sum_{i=0}^{\alpha} |S_i| \geq \alpha + 1$, for every $\alpha \in [1 : T]$. We use induction on α. If $\sum_{i=0}^{1} |S_i| < 2$, then $|S_0| + |S_1| = 1$. Thus, $S_j = \emptyset, \forall j \geq 1$. That implies that $\sum_{i=0}^{T} |S_i| = 1$, which is a contradiction. Let now $1 < \alpha \leq T$ and assume by induction that $\sum_{i=0}^{\alpha-1} |S_i| \geq \alpha$. Since $\sum_{i=0}^{\alpha-1} |S_i| \leq \sum_{i=0}^{\alpha} |S_i|$. Thus, $\sum_{i=0}^{\alpha} |S_i| \geq \sum_{i=0}^{\alpha-1} |S_i| \geq \alpha$. If $\sum_{i=0}^{\alpha} |S_i| < \alpha + 1$, then $\sum_{i=0}^{\alpha} |S_i| = \sum_{i=0}^{\alpha-1} |S_i| = \alpha$. Hence, $S_\alpha = \emptyset$, and $\sum_{i=0}^{T} |S_i| = \sum_{i=0}^{\alpha-1} |S_i| = k^n$. This contradicts the minimality of T. Thus $\sum_{i=0}^{\alpha} |S_i| \geq \alpha + 1$.

From the above discussion, we have $k^n = \sum_{i=0}^{T} |S_i| \geq T + 1$, and therefore, $T \leq k^n - 1$.

(Sufficiency) The proof of sufficiency is constructive. Assume that there exists an integer $T \in [1 : k^n - 1]$ such that $\sum_{i=0}^{T} |S_i| = k^n$. Then $\bigcup_{i=0}^{T} S_i = \Delta_{k^n}$. For an arbitrary given initial strategy profile $x(t_0) = \delta_{k^n}^{j} \in \{x(-\tau), x(-\tau+1), \cdots, x(0)\} \subseteq \Delta_{k^n}$, either $x(t_0) \in S_0$ or $x(t_0) \in \bigcup_{i=1}^{T} \widehat{S}_i$ or $x(t_0) \in \bigcup_{i=1}^{T} \widetilde{S}_i$.

Let the feedback control matrix be $H = \delta_{k^m+1}[\nu_1 \ \nu_2 \ \cdots \ \nu_{k^n}]$, where each column of matrix H can be designed as follows:

(i) If $\delta_{k^n}^{j} \in S_0, j \in [1 : k^n]$, then

$$\nu_j = k^m + 1.$$

(ii) If $\delta_{k^n}^j \in \bigcup_{i=1}^T \widehat{S}_i, j \in [1:k^n]$, then

$$\nu_j \in P(\nu_j),$$

where $P(\nu_j) = \{\nu_j | \mathrm{Col}_{\nu_j}(\mathrm{Blk}_j(\widehat{L})) \in S_{i-1}\}$.

(iii) If $\delta_{k^n}^j \in \bigcup_{i=1}^T \widetilde{S}_i, j \in [1:k^n]$, then

$$\nu_j = k^m + 1.$$

Case 1: $x(t_0) \in S_0 = E_r(\delta_{k^n}^\theta)$. From the construction of H, there exists a positive integer $r_0 \in [1:k^n]$, such that

$$\begin{aligned} x(t_0 + r_0(\tau+1)) &= Lx(t_0 + (r_0 - 1)(\tau+1)) \\ &= L^{r_0}\delta_{k^n}^j \\ &= \delta_{k^n}^\theta. \end{aligned}$$

Hence, from condition (6.8), we have $x(t_0 + k(\tau+1)) = \delta_{k^n}^\theta, \forall k \geq r_0$ and $k \in \mathbb{N}^+$.

Case 2: $x(t_0) \in \bigcup_{i=1}^T \widehat{S}_i$. Without loss of generality, let $x(t_0) \in \widehat{S}_{i_0}$, then

$$\begin{aligned} x(t_0 + (\tau+1)) &= \overline{L}u(t_0)x(t_0) \\ &= \overline{L}W_{[k^n,k^m]}x(t_0)u(t_0) \\ &= \mathrm{Blk}_j(\widehat{L})u(t_0) \\ &= \mathrm{Col}_{\nu_i}(\mathrm{Blk}_j(\widehat{L})) \in S_{i_0-1}. \end{aligned}$$

If $i_0 = 1$, then $x(t_0 + (\tau+1)) \in S_0$. From the construction of S_0, there exists an integer $r_1 \in [2:k^n]$, such that $x(t_0 + r_1(\tau+1)) = \delta_{k^n}^\theta$. Hence, $x(t_0 + k(\tau+1)) = \delta_{k^n}^\theta, \forall k \geq r_1$ and $k \in \mathbb{Z}^+$.

Otherwise, let $i_0 > 1$. Since $S_{i_0-1} = \widehat{S}_{i_0-1} \bigcup \widetilde{S}_{i_0-1}$, we get that if $x(t_0+(\tau+1)) \in \widehat{S}_{i_0-1}$, then $x(t_0 + 2(\tau+1)) \in S_{i_0-2}$; otherwise, if $x(t_0 + (\tau+1)) \in \widetilde{S}_{i_0-1}$, then there exists a positive integer r_2, such that $x(t_0 + r_2(\tau+1)) \in \widehat{S}_{i_0-1}$. Repeating this procedure, we can find a time $\widehat{T} \leq T$, such that $x(t_0 + \widehat{T}(\tau+1)) \in S_0$. Based on the analysis above, we know that $x(t_0)$ will eventually converge to $\delta_{k^n}^\theta$.

Case 3: $x(t_0) \in \bigcup_{i=1}^T \widetilde{S}_i$. Without loss of generality, let $x(t_0) \in \widetilde{S}_{i_0}$. Then $x(t_0)$ can evolve to $\delta_{k^n}^\theta$ naturally in finite steps. Based on the above discussion, the dynamics will converge to the strategy profile $x^* = \delta_{k^n}^\theta$ and maintain the desired profile unchanged.

From the arbitrariness of initial strategy profiles $x(t_0) \in \{x(-\tau), x(-\tau+1), \cdots, x(0)\}$, we obtain that $x^* = \delta_{k^n}^\theta$ must be reachable from any initial strategy profiles $x(-\tau), x(-\tau+1), \cdots, x(0) \in \Delta_{k^n}$. Moreover, $x^* = \delta_{k^n}^\theta$ is the fixed point of (6.5). Therefore, the evolutionary dynamics can converge to the desired only final strategy profile globally. □

The following corollary immediately follows from Theorem 6.3.3.

Corollary 6.3.4. *If* $\bigcup_{i=1}^{T} \widehat{S}_i = \{\widehat{\delta_{k^n}^{i_1}}, \widehat{\delta_{k^n}^{i_2}}, \cdots, \widehat{\delta_{k^n}^{i_l}}\}$, *the total number of different state feedback event-triggered controller which can make the evolutionary dynamics converge to the desired only final strategy profile globally is* $\prod_{j=1}^{l} |P(\nu_j)|$.

Remark 6.3.1. *The proof of sufficiency in Theorem 6.3.3 provides a method for constructing all the valid state feedback event-triggered controllers of the evolutionary dynamics of DNEGs. The proposed algorithm is not sensitive to system parameters. Here, we concentrate our attention to the theory behind the method. In the future, we will apply these results to the practical application of the method for given network.*

Remark 6.3.2. *In our model all players have the same time delay* $\tau + 1$. *In this situation, the time delay only affects the convergence time of the system. For the system with different time delays, the time delays will affect both the stabilization and the convergence time. We plan to investigate this kind of system in future works.*

6.4 An illustrative example

In this section, we provide an illustrative example to show how to use the results proposed in the previous sections to study the global convergence problem of DNEGs.

Example 6.4.1. *Consider the following DNEG, which has the following items:*

- $N = \{1, 2, 3, 4\}$ *is the player set; and each player has the same strategy set* $S = \{1, 2\}$. *The network topological structure for the four players is shown in Figure 6.2.*

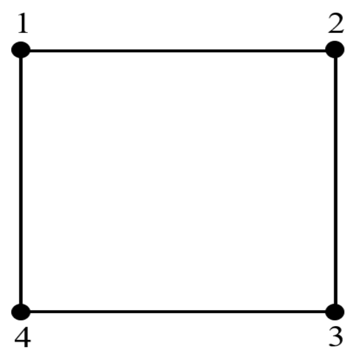

Figure 6.2: The network graph of Example 6.4.1

- *the payoff bi-matrix of the FNG is shown in Table 6.1.*
- *the strategy updating rule is the Parallel MBRAR.*

Table 6.1: Payoff bi-matrix of DNEG

$P_1 \setminus P_2$	1	2
1	(1, 2)	(0, 0)
2	(0, 0)	(2, 1)

Assume that the time-invariant delay is $\tau + 1 = 2$. Then the algebraic form of the evolutionary dynamics for each player can be obtained as follows.

$$\begin{aligned} x_1(t+1) &= f_1(x_4(t-1), x_1(t-1), x_2(t-1)) = Mx_4(t-1)x_1(t-1)x_2(t-1), \\ x_2(t+1) &= f_2(x_1(t-1), x_2(t-1), x_3(t-1)) = Mx_1(t-1)x_2(t-1)x_3(t-1), \\ x_3(t+1) &= f_3(x_2(t-1), x_3(t-1), x_4(t-1)) = Mx_2(t-1)x_3(t-1)x_4(t-1), \\ x_4(t+1) &= f_4(x_3(t-1), x_4(t-1), x_1(t-1)) = Mx_3(t-1)x_4(t-1)x_1(t-1), \end{aligned}$$

where $M = \delta_2[1\ 2\ 1\ 2\ 2\ 2\ 2\ 2]$.

Then, based on Lemma 1.1.2, the algebraic form above can be transformed into

$$x_i(t+1) = M_i x(t-1),\ i \in [1:4],$$

where

$$\begin{aligned} M_1 &= MD_f^{2^3,2} W_{[2^3,2]}, \\ M_2 &= MD_f^{2^3,2}, \\ M_3 &= MD_r^{2,2^3}, \\ M_4 &= MD_f^{2^3,2} W_{[2^2,2^2]}. \end{aligned}$$

Finally, the evolutionary dynamics of the whole network can be converted into the following algebraic formulation:

$$x(t+1) = Lx(t-1), \tag{6.16}$$

where $L = \delta_{16}[1\ 11\ 6\ 16\ 11\ 11\ 16\ 14\ 6\ 16\ 6\ 15\ 16\ 8\ 12\ 1]$.

It is obvious that there are a fixed point δ_{16}^1 and two cycles $\{\delta_{16}^6, \delta_{16}^{11}\}, \{\delta_{16}^{12}, \delta_{16}^{15}\}$ with length two. For more details about the fixed point and cycle of NEGs, please refer to [14].

Next, we study the evolutionary dynamics of the DNEGs that ensure globally convergence to the desired only final strategy profile $x^* = \delta_{16}^1$ under state feedback event-triggered control. In the following, suppose that Player 1 is affected by an external player, and Player 1 imitates the strategy of controller unconditionally when the event is triggered, while the other players will choose strategies based on the original updating rule.

Let the strategy of controller be $1 \sim \delta_3^1, 2 \sim \delta_3^2$ while the no control action is $u_0 \sim \delta_3^3$. The algebraic form of the game under event-triggered state feedback control is

$$x(t+1) = [\overline{L}\ \ L]\widetilde{u}(t-1)x(t-1), \tag{6.17}$$

where $\widetilde{u}(t-1) \in \Delta_3$, $x(t-1) = \ltimes_{i=1}^{4} x_i(t-1) \in \Delta_{2^4}$, and

$$\begin{aligned}\overline{L} &= \delta_{16}[1\ \ 3\ \ 6\ \ 8\ 3\ 3\ \ 8\ 6\ 6\ 8\ 6\ \ 7\ 8\ 8\ \ 4\ 1\\ &\qquad 9\ 11\ 14\ 16\ 11\ 11\ 16\ 14\ 14\ 16\ 14\ 15\ 16\ 16\ 12\ 9].\end{aligned}$$

Then, we aim to find out all the valid feedback control matrices.

First, we split the matrix $\widehat{L} = \overline{L}W_{[2^4,2]}$ into 2^4 blocks:

$$\mathrm{Blk}_1(\widehat{L}) = \delta_{16}[1\ 9],\ \mathrm{Blk}_2(\widehat{L}) = \delta_{16}[3\ 11],\ \mathrm{Blk}_3(\widehat{L}) = \delta_{16}[6\ 14],\ \mathrm{Blk}_4(\widehat{L}) = \delta_{16}[8\ 16],$$

$$\mathrm{Blk}_5(\widehat{L}) = \delta_{16}[3\ 11],\ \mathrm{Blk}_6(\widehat{L}) = \delta_{16}[3\ 11],\ \mathrm{Blk}_7(\widehat{L}) = \delta_{16}[8\ 16],\ \mathrm{Blk}_8(\widehat{L}) = \delta_{16}[6\ 14],$$

$$\mathrm{Blk}_9(\widehat{L}) = \delta_{16}[6\ 14],\ \mathrm{Blk}_{10}(\widehat{L}) = \delta_{16}[8\ 16],\ \mathrm{Blk}_{11}(\widehat{L}) = \delta_{16}[6\ 14],\ \mathrm{Blk}_{12}(\widehat{L}) = \delta_{16}[7\ 15],$$

$$\mathrm{Blk}_{13}(\widehat{L}) = \delta_{16}[8\ 16],\ \mathrm{Blk}_{14}(\widehat{L}) = \delta_{16}[8\ 16],\ \mathrm{Blk}_{15}(\widehat{L}) = \delta_{16}[4\ 12],\ \mathrm{Blk}_{16}(\widehat{L}) = \delta_{16}[1\ 9].$$

From the constructing procedure of $S_i, i \geq 0$, we have

$$S_0 = E_r(\delta_{16}^1) = \{\delta_{16}^1, \delta_{16}^4, \delta_{16}^7, \delta_{16}^{10}, \delta_{16}^{13}, \delta_{16}^{16}\},$$

$$\widehat{S}_1 = \widehat{E}(S_0)\bigcap[\Delta_{2^4} \setminus S_0] = \{\delta_{16}^{12}, \delta_{16}^{14}, \delta_{16}^{15}\}, \quad \widetilde{S}_1 = E_r(\widehat{S}_1) \setminus \widehat{S}_1 = \{\delta_{16}^8\},$$

$$\widehat{S}_2 = \widehat{E}(S_1)\bigcap[\Delta_{2^4} \setminus (S_0 \cup S_1)] = \{\delta_{16}^3, \delta_{16}^9, \delta_{16}^{11}\}, \quad \widetilde{S}_2 = E_r(\widehat{S}_2) \setminus \widehat{S}_2 = \{\delta_{16}^2, \delta_{16}^5, \delta_{16}^6\}.$$

Thus, for $T = 2$, $\sum_{i=0}^{2} |S_i| = 2^4$.

Based on the above analysis, we have $\Gamma = \widehat{S}_1 \cup \widehat{S}_2 = \{\delta_{16}^3, \delta_{16}^9, \delta_{16}^{11}, \delta_{16}^{12}, \delta_{16}^{14}, \delta_{16}^{15}\}$, and $P(\nu_3) = \{\delta_3^2\}$, $P(\nu_9) = \{\delta_3^2\}$, $P(\nu_{11}) = \{\delta_3^2\}$, $P(\nu_{12}) = \{\delta_3^1\}$, $P(\nu_{14}) = \{\delta_3^2\}$, $P(\nu_{15}) = \{\delta_3^1\}$. Thus, the only feedback control matrix $H \in \mathcal{L}_{3\times 2^4}$ is given by

$$H = \delta_3[3\ 3\ 2\ 3\ 3\ 3\ 3\ 3\ 2\ 3\ 2\ 1\ 3\ 2\ 1\ 3].$$

We consider the algorithm proposed in [74] to design traditional state feedback controller. Then, we can obtain a sequence of sets as follows

$$E_1^{'}(\delta_{16}^1) = \{\delta_{16}^1, \delta_{16}^{16}\},\ E_2^{'}(\delta_{16}^1) = \{\delta_{16}^1, \delta_{16}^4, \delta_{16}^7, \delta_{16}^{10}, \delta_{16}^{13}, \delta_{16}^{14}, \delta_{16}^{16}\},$$

$$E_3^{'}(\delta_{16}^1) = \{\delta_{16}^1, \delta_{16}^3, \delta_{16}^4, \delta_{16}^7, \delta_{16}^8, \delta_{16}^9, \delta_{16}^{10}, \delta_{16}^{11}, \delta_{16}^{12}, \delta_{16}^{13}, \delta_{16}^{14}, \delta_{16}^{15}, \delta_{16}^{16}\},\ E_4^{'}(\delta_{16}^1) = \Delta_{16},$$

where $E_r^{'}(\delta_{16}^1)$, $r \in [1:4]$ denotes the set consisting of all the initial strategy profiles which can be steered to x^* in r steps by a control sequence $u(0), u(1), \cdots, u(r-1)$. Therefore, the control need to be executed at least 4 times to achieve global convergence. Moreover, after the system is stable, a constant control should be executed all the time to maintain the stabilization of the system. In the method proposed here, the control only need to be executed 2 times to achieve the same target, and a constant control is not necessary to maintain the performance of the system considering that x^* is a fixed point and it will not trigger the control any more. It is no doubt that the event-triggered state feedback control can greatly save the control costs.

Finally, the matrix Φ is obtained as

$$\Phi = \delta_{256}[1\ 161\ 209\ 241\ \cdots\ 256\ 256\ 64\ 16].$$

By calculation, there exists $T = 8$, such that

$$\mathrm{Col}(\Phi^8) = \{\ltimes_{i=1}^{2} \delta_{2^4}^1\}.$$

Therefore, the evolutionary dynamics can converge to the desired strategy profile δ_{16}^1 globally under state feedback event-triggered controller $\widetilde{u}(t) = Hx(t)$.

6.5 Conclusions

In this chapter, dynamics and control problems for a class of DNEGs have been investigated. The dynamics of the DNEGs have been first converted into algebraic forms via the STP. Using the algebraic forms, the dynamic behaviors of the DNEGs have been discussed. Based on that, all valid state feedback event-triggered controllers are constructed to affect the evolution of the game, and necessary and sufficient conditions are derived to assure the global convergence of the desired only strategy profile of DNEGs. Moreover, the number of all valid state feedback event-triggered controllers is obtained. An illustrative example has been presented to show the effectiveness of the obtained results.

Chapter 7

Algebraization and Optimization of Networked Evolutionary Boxed Pig Games with Passive Reward and Punishment

7.1 Introduction

The boxed pig game was firstly put forward by John Nash in [101]. As we all know, the free-rider phenomenon of small pig is prevalent in the reality. For example, the retailers wait for the bankers take the action called “the sedan chair” in stock market or the employees do not create benefit but share the results in the enterprise. In reaction to this phenomenon, early studies mainly focused on the areas such as economics and enterprise talent management, *etc*, and scholars have analyzed the superiority of the mechanism of passive reward and punishment in promoting the player’s cooperation behavior in theory [76, 96]. Due to the lack of effective mathematical tools, it was hard to systematically analyze the influence of passive reward and punishment on the final cooperation level of the whole network until the emergence of STP.

Inspired by the successful applications of STP method in the analysis and control of logical dynamic systems [13, 14, 69, 65] and control and strategy optimization of evolutionary games [32, 139, 29, 34], this chapter investigates the impact of the passive reward and punishment on the evolutionary dynamics of networked evolutionary boxed pig games. The main contributions of this chapter are:

- The model of the boxed pig game with the passive reward and punishment is investigated, and an algorithm to calculate the algebraic form of evolutionary dynamics is constructed by using STP method.
- The impact of the reward and punishment parameters on the final cooperation level of the boxed pig game is discussed.

7.2 Model description

In this section, the model of the boxed pig game with the mechanism of passive reward and punishment is introduced.

First, the traditional boxed pig game [108] can be described as follows.

There are a small pig and a big pig in the pigsty. On one side of the pigsty there is a food storage vessel, and on the other side there is a pedal to control the supply of food. When the pedal is pressed, a certain amount of food will be turned into the vessel, while a certain amount of food will be expended during the process of pressing the pedal. Then the payoff bi-matrix for the traditional boxed pig game is shown in Table 7.1, where

Table 7.1: Payoff bi-matrix of traditional boxed pig game

$P_1 \setminus P_2$	$C = 1$	$D = 2$
$C = 1$	(a, b)	(m, s)
$D = 2$	(e, f)	(g, h)

$e > a > m = s > b > 0 > f$, and $g = h = 0$, P_1 and P_2 represent the big pig and the small pig respectively, C means press the pedal and D means wait.

It is easy to know that the payoff matrices of big pig and small pig are:

$$\begin{bmatrix} a & m \\ e & g \end{bmatrix}, \qquad \begin{bmatrix} b & s \\ f & h \end{bmatrix}.$$

Through the payoff bi-matrix, it is enough to verify that (C, D) is a Nash equilibrium of the game, *i.e.* big pig will press the pedal and small pig will wait. That is, small pig will choose the strategy means “free-rider”, while big pig will travel between pedal and vessel constantly. It is obvious that both small pig and big pig do not give full play to their motivation during the gambling process, and it is contrary to the fairness of the game.

Then, we consider optimizing the original model according to the following manner.

We assume that someone (such as feeder) who can be contacted by all players will apply to cooperator and defector reward and punishment respectively: for a player taking part in the game, if he chooses cooperation, the feeder will provide to him a certain amount of food as a reward for his cooperative behavior; on the contrary, the feeder will decrease the amount of food if the player chooses defection. Under the mechanism of reward and punishment, the payoff bi-matrix is shown in Table 7.2, where α and β represent the amount

Table 7.2: Payoff bi-matrix under reward and punishment

$P_1 \setminus P_2$	$C = 1$	$D = 2$
$C = 1$	$(a+\alpha, b+\alpha)$	$(m+\alpha, s-\beta)$
$D = 2$	$(e-\beta, f+\alpha)$	$(g-\beta, h-\beta)$

of reward and punishment applied by the feeder to cooperator and defector, respectively. It is obvious that $\alpha \geq 0$, $\beta \geq 0$. Thus, the payoff matrices of big pig and small pig can be shown in the following form respectively:

$$A = \begin{bmatrix} a+\alpha & m+\alpha \\ e-\beta & g-\beta \end{bmatrix},$$

$$B = \begin{bmatrix} b+\alpha & s-\beta \\ f+\alpha & h-\beta \end{bmatrix}.$$

We know that if $(i, j) \in \mathcal{E}$, then big pig and small pig play fundamental network game and their payoffs can be calculated as follows, respectively:

$$c_i(t) = V_r^T(A)x_i(t)x_j(t), \tag{7.1}$$

$$\begin{aligned} c_j(t) &= V_r^T(B)x_i(t)x_j(t) \\ &= V_r^T(B)W_{[2]}x_j(t)x_i(t), \end{aligned} \tag{7.2}$$

where $c_i(t)$ and $c_j(t)$ represent the payoffs of big pig and small pig at time t respectively; $x_i(t)$ and $x_j(t)$ represent the strategies of big pig and small pig at time t respectively .

Remark 7.2.1. *There must be some constraints on the reward α and punishment β, which depends on particular models. In our model, after the reward, the quantity of total food should be less than certain upper limitation, e.g., vessel capacity; and punish quantity should be less than food available in the vessel.*

7.3 Algebraic formulation of evolutionary dynamic process

In the following, we consider the algebraic formulation and optimization problem of networked evolutionary boxed pig game with the mechanism of passive reward and punishment.

A networked evolutionary boxed pig game with multi-players is denoted as $((N, \mathcal{E}), G, \Pi)$, where

- $N = \{1, 2, \cdots, n\}$ is the set of players, and each player may be small pig or big pig. $\mathcal{E}$ is a directed cycle network graph with n nodes to depict the positional relationship among players. For any $(i, j) \in \mathcal{E}$, i represents the big pig and j represents the small pig. Let its adjacency matrix be

$$\begin{aligned} E &= (e_{ij})_{n\times n} \\ &= \begin{bmatrix} 0 & e_{12} & 0 & 0 & \cdots & 0 & e_{1n} \\ e_{21} & 0 & e_{23} & 0 & \cdots & 0 & 0 \\ 0 & e_{32} & 0 & e_{34} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & e_{n-1,n} \\ e_{n1} & 0 & 0 & 0 & \cdots & e_{n,n-1} & 0 \end{bmatrix}, \end{aligned}$$

  where $e_{ij} \in \{1, 0\}$, and $e_{ij} = 1$ if and only if $(i, j) \in \mathcal{E}$. Obviously $e_{ij} \wedge e_{ji} = 0$.

- G is the fundamental network game, *i.e.*, the boxed pig game with two players under passive reward and punishment.

- Π is the local information based strategy updating rule.

First, the algebraic form of the evolutionary dynamics of the game is investigated.

Denoting the strategy of player i at time t by $x_i(t)$, his local information based strategy updating rule can be expressed with a set of mappings as follows :

$$x_i(t+1) = f_i\Big(\{x_j(t), c_j(t) | j \in U(i)\}\Big),\ t \geq 0, i \in N, \tag{7.3}$$

where $U(i)$ represents the set of neighborhood players of i. Moreover, the average payoff [18] of player i is

$$c_i(t) = \frac{1}{|U(i)| - 1} \sum_{j \in U(i)\setminus\{i\}} c_{ij}(t), i \in N,$$

where $c_{ij} : S \times S \to R$ is the payoff of player i playing with his neighbor j, $S = \{1, 2\}$ is the set of strategies and 1, 2 represents cooperation and defection respectively.

The strategy updating rule considered in this chapter is UISUR [18] with fixed priority: the strategy of player i at time $t+1$, $x_i(t+1)$, is selected as the best strategy from strategies of neighborhood players $j \in U(i)$ at time t. Precisely, if

$$j^\star = \arg\max_{j\in U(i)} c_j(x(t)), \tag{7.4}$$

then

$$x_i(t+1) = x_{j^\star}(t). \tag{7.5}$$

When the players with best payoff are not unique, *i.e.*,

$$\arg\max_{j\in U(i)} c_j(x(t)) := \{j_1^\star, j_2^\star, \cdots, j_r^\star\}, \tag{7.6}$$

then we may choose one corresponding to a priority as

$$j^\star = \min\Big\{\mu | \mu \in \arg\max_{j\in U(i)} c_j(x(t))\Big\}. \tag{7.7}$$

To construct the algebraic form of the game, one can take the following two key steps:

(i) convert the average payoff function of each player $i \in [1:n]$ into an algebraic form;

(ii) construct the algebraic form for the evolutionary dynamic of each player i.

For step (i), since the network graph is a directed cycle with n nodes, we identify $x_0 = x_n, x_{-1} = x_{n-1}$. Using the vector form of logical variables, let $1 \sim \delta_2^1,\ 2 \sim \delta_2^2$, then $S \sim \Delta$

Thus, the algebraic form of average payoff for any player $i \in [1:n]$ is given below

$$\begin{aligned}
c_i(t) &= c_i(x_i(t), x_j(t) \mid j \in U(i)) \\
&= \frac{1}{|U(i)|-1} \sum_{j\in U(i)\setminus\{i\}} \Big(e_{ij} V_r^T(A) x_i(t) x_j(t) + e_{ji} V_r^T(B) W_{[2]} x_i(t) x_j(t)\Big) \\
&= \frac{1}{2}\Big[(e_{i,i+1} V_r^T(A) x_i(t) x_{i+1}(t) + e_{i+1,i} V_r^T(B) W_{[2]} x_i(t) x_{i+1}(t)) \\
&\quad + (e_{i,i-1} V_r^T(A) x_i(t) x_{i-1}(t) + e_{i-1,i} V_r^T(B) W_{[2]} x_i(t) x_{i-1}(t))\Big] \\
&= \frac{1}{2}\Big[(e_{i,i+1} V_r^T(A) + e_{i+1,i} V_r^T(B) W_{[2]}) D_f^{2^2,2^{n-2}} W_{[2^{i-1},2^{n-i+1}]} \\
&\quad + (e_{i,i-1} V_r^T(A) + e_{i-1,i} V_r^T(B) W_{[2]}) W_{[2]} D_r^{2^{n-2},2^2} W_{[2^i,2^{n-i}]}\Big] x(t) \\
&:= P_i x(t),
\end{aligned} \tag{7.8}$$

where

$$P_i = \frac{1}{2}\Big[(e_{i,i+1} V_r^T(A) + e_{i+1,i} V_r^T(B) W_{[2]}) D_f^{2^2,2^{n-2}} W_{[2^{i-1},2^{n-i+1}]}$$

$$+(e_{i,i-1}V_r^T(A)+e_{i-1,i}V_r^T(B)W_{[2]})W_{[2]}D_r^{2^{n-2},2^2}W_{[2^i,2^{n-i}]}\Big]\in\mathbb{R}_{1\times 2^n}$$

is the structural matrix of average payoff c_i, $\text{Col}_k(P_i)$, $k\in[1:2^n]$ represents the benefit of player i under the profile $\delta_{2^n}^k$. $x_i(t)\in\Delta$ is the strategy of player i at time t, and $x(t)=\ltimes_{i=1}^n x_i(t)\in\ \Delta_{2^n}$.

Since the network graph is a cycle, then for player $i\in[1:n]$, according to the strategy updating rule, to obtain the strategy of player i at time t, we need to compare average payoffs of player $i-1, i+1$, and i at time t.

First, if $x(t)=\delta_{2^n}^k$, $k\in[1:2^n]$, then the strategy of the $i-$th player can be calculated as follows:

$$x_i(t)=D_f^{2,2^{n-1}}W_{[2^{i-1},2]}x(t). \tag{7.9}$$

Let now

$$H_i^k = [P_{i-1}\delta_{2^n}^k,\ P_i\delta_{2^n}^k,\ P_{i+1}\delta_{2^n}^k], \tag{7.10}$$

$$F_i^k = [D_f^{2,2^{n-1}}W_{[2^{i-2},2]}\delta_{2^n}^k,\ D_f^{2,2^{n-1}}W_{[2^{i-1},2]}\delta_{2^n}^k, D_f^{2,2^{n-1}}W_{[2^i,2]}\delta_{2^n}^k]. \tag{7.11}$$

Thus, the problem of identifying the one who has the maximal average payoff among players $i-1,i+1$ and i, can be converted into finding the column index l_k^i, $k\in[1:2^n]$, such that

$$\text{Col}_{l_k^i}(H_i^k)\geq \text{Col}_j(H_i^k),\quad j\in[1:3]. \tag{7.12}$$

If l_k^i is not unique, one can pick out the unique column index according to the priority:

$$\varepsilon_k^i=\min\left\{l_k^i|\text{Col}_{l_k^i}(H_i^k)\geq \text{Col}_j(H_i^k)\right\}, \tag{7.13}$$

where $j=1,2,3$, then the strategy of player i at time $t+1$, can be chosen as $x_i(t+1)=\text{Col}_{\varepsilon_k^i}(F_i^k)$.

Next, we identify $\text{Col}_k(L_i)=\text{Col}_{\varepsilon_k^i}(F_i^k), k\in[1:2^n]$, for each player i, and we can obtain the evolutionary dynamic equation as the following algebraic form

$$x_i(t+1)=L_ix(t), \tag{7.14}$$

where $L_i\in\mathcal{L}_{2\times 2^n}$ and $x(t)=\ltimes_{i=1}^n x_i(t)\in\Delta_{2^n}$.

Based on the above analysis, we have the following algorithm to construct the algebraic form for the networked evolutionary boxed pig games with passive reward and punishment.

Algorithm 9 Constructing the algebraic formulation of the game:

Step 1: Calculate the structural matrix, P_i, of the average payoff function for each player $i \in [1:n]$,

$$\begin{aligned} P_i &= \frac{1}{2}\Big[(e_{i,i+1}V_r^T(A) + e_{i+1,i}V_r^T(B)W_{[2]})D_f^{2^2,2^{n-2}}W_{[2^{i-1},2^{n-i+1}]} \\ &\quad +(e_{i,i-1}V_r^T(A) + e_{i-1,i}V_r^T(B)W_{[2]})W_{[2]}D_r^{2^{n-2},2^2}W_{[2^i,2^{n-i}]}\Big]. \end{aligned}$$

Step 2: Calculate matrices

$$\begin{aligned} H_i^k &= [P_{i-1}\delta_{2^n}^k,\ P_i\delta_{2^n}^k,\ P_{i+1}\delta_{2^n}^k], \\ F_i^k &= [D_f^{2,2^{n-1}}W_{[2^{i-2},2]}\delta_{2^n}^k, D_f^{2,2^{n-1}}W_{[2^{i-1},2]}\delta_{2^n}^k, D_f^{2,2^{n-1}}W_{[2^i,2]}\delta_{2^n}^k]. \end{aligned}$$

For all $k \in [1:2^n]$, find the column index ε_k^i, such that

$$\varepsilon_k^i = \min\Big\{l_k^i|\mathrm{Col}_{l_k^i}(H_i^k) \geq \mathrm{Col}_j(H_i^k), j \in [1:3]\Big\},$$

and let $\mathrm{Col}_k(L_i) = \mathrm{Col}_{\varepsilon_k^i}(F_i^k)$.

Step 3: Construct the algebraic form of the game under study as

$$x(t+1) = Lx(t), \tag{7.15}$$

where $\mathrm{Col}_i(L) = \mathrm{Col}_i(L_1) \ltimes \mathrm{Col}_i(L_2) \ltimes \cdots \ltimes \mathrm{Col}_i(L_n)$, $i \in [1:n]$, and $L \in \mathcal{L}_{2^n \times 2^n}$.

Based on Algorithm 9, the evolutionary dynamics of networked evolutionary boxed pig game can be equivalently formulated as a BN. Thus, the discussion about dynamic analysis and optimization of the game can be resorted to the results of analysis and control of BNs. Then, according to the strategy updating rule, we have

$$\mathrm{Col}_1(L) = \delta_{2^n}^1 = \delta_2^1\delta_2^1\cdots\delta_2^1,$$

$$\mathrm{Col}_{2^n}(L) = \delta_{2^n}^{2^n} = \delta_2^2\delta_2^2\cdots\delta_2^2,$$

i.e. $\delta_{2^n}^1$ and $\delta_{2^n}^{2^n}$ are two fixed points of the system, which shows that full cooperation and full defection are two stable profiles of the given networked evolutionary boxed pig game.

Remark 7.3.1. *Among all players $\{1,2,\cdots,n\}$, if a player chooses cooperation as his initial state, the full defection profile will not appear during the dynamic process.*

Remark 7.3.2. *It must be noted that we just considered a simple cycle network in this chapter, and the construction of the algebraic form and the following main results can be easily extended to other more general networks.*

7.4 Optimization of networked evolutionary boxed pig games

In this section, we provide a necessary and sufficient condition to detect whether the final dynamic behavior of boxed pig game with the mechanism of passive reward and punishment can converge to full cooperation profile, and discuss how to adjust the values of reward and punishment parameters α, β, such that all initial profiles except full defection can converge to a full cooperation profile.

Theorem 7.4.1. *Consider the passive reward and punishment networked evolutionary boxed pig game with the algebraic form (7.15). The initial profile $x(0) = \delta_{2^n}^i$ converges to full cooperation profile, if and only if there exists a time $T \in [0 : 2^n]$, such that*

$$\mathrm{Col}_i(L^T) = \delta_{2^n}^1, \ i \in [1 : 2^n]. \tag{7.16}$$

Proof. For any initial profile $x(0) = \delta_{2^n}^i$, the dynamics of system (7.15) can be expressed as

$$\begin{aligned} x(1) &= Lx(0), \\ x(2) &= Lx(1) \\ &= L^2 x(0), \\ &\cdots \\ x(t) &= Lx(t-1) \\ &= \cdots \\ &= L^t x(0) \\ &= L^t \delta_{2^n}^i \\ &= \mathrm{Col}_i(L^t). \end{aligned}$$

(Necessity) We prove it by contradiction. Assume for any time t, $\mathrm{Col}_i(L^t) \neq \delta_{2^n}^1$. Then we have $\mathrm{Col}_i(L^t) = L^t \delta_{2^n}^i = L^t x(0) = x(t) \neq \delta_{2^n}^1$. That contradicts the fact that $x(0)$ converges to full cooperation profile $x_e = \delta_{2^n}^1$. Hence (7.16) is satisfied.

(Sufficiency) If there exists a time T, such that $\mathrm{Col}_i(L^T) = \delta_{2^n}^1$ holds, then for any $\forall t \geq T$, we have

$$\begin{aligned} x(t) &= L^t x(0) \\ &= L^{t-T} L^T x(0) \\ &= L^{t-T} \mathrm{Col}_i(L^T) \\ &= L^{t-T} \delta_{2^n}^1. \end{aligned}$$

Since $\delta_{2^n}^1$ is the fixed point of system (7.15), we know that $\delta_{2^n}^1 = L\delta_{2^n}^1$, and $x(t) = L^{t-T}\delta_{2^n}^1 = \delta_{2^n}^1$. Therefore, $x(0) = \delta_{2^n}^i$ converges to full cooperation profile. □

Remark 7.4.1. *Let*

$$\Gamma_e = \left\{\delta_{2^n}^i | \mathrm{Col}_i(L^{2^n}) = \delta_{2^n}^1\right\}. \tag{7.17}$$

Since there are 2^n different profiles, for any initial profile $x(0) = \delta_{2^n}^i$, $i \in [1 : 2^n]$, if the initial profile $x(0) = \delta_{2^n}^i$ can not reach to $x_e = \delta_{2^n}^1$ after time 2^n, it will not reach to full cooperation any more. Conversely, if it reaches to the profile $x_e = \delta_{2^n}^1$ within time 2^n, it will maintain full cooperation $x_e = \delta_{2^n}^1$ unchanged. Then, if $\mathrm{Col}_i(L^{2^n}) = \delta_{2^n}^1$, we have $x(t) = L^t\delta_{2^n}^i = L^{t-2^n}L^{2^n}\delta_{2^n}^i = L^{t-2^n}\delta_{2^n}^1 = \delta_{2^n}^1$, for all $t \geq 2^n$, that is, the initial profile $x(0) = \delta_{2^n}^i$, $i \in [1 : 2^n]$, can converge to full cooperation. Then, $x(0)$ converges to full cooperation profile x_e, if and only if $x(0) \in \Gamma_e$. We call Γ_e the convergence domain of the game. If we denote the number of elements in convergence domain Γ_e as $|\Gamma_e|$, it is obvious that $|\Gamma_e| \leq 2^n - 1$.

The optimization goal is to maximize the number of elements in Γ_e. In the following, we discuss how to adjust the values of reward and punishment parameters α and β, such that $|\Gamma_e| = 2^n - 1$.

According to UISUR, to make $|\Gamma_e| = 2^n - 1$, we just need to adjust the values of α and β, such that the average payoff of defector is lower than the average payoff of cooperator. Then it will follow that the defector will imitate the strategy of cooperator.

For any player $i \in [1 : n]$, according to the strategy updating rule, first we need to calculate the average payoff of player i and his neighbors $i-1$ and $i+1$. Those three payoffs, are completely determined by $U_2(i)$. Thus, the average payoff of player i can be calculated as follows :

$$\begin{aligned}
c_i(t) &= \frac{1}{|U(i)|-1}\sum_{j\in U(i)\setminus\{i\}}\Big(e_{ij}V_r^T(A)x_i(t)x_j(t) + e_{ji}V_r^T(B)W_{[2]}x_i(t)x_j(t)\Big)\\
&= \frac{1}{2}\Big[(e_{i,i+1}V_r^T(A)x_i(t)x_{i+1}(t) + e_{i+1,i}V_r^T(B)W_{[2]}x_i(t)x_{i+1}(t))\\
&\quad +(e_{i,i-1}V_r^T(A)x_i(t)x_{i-1}(t) + e_{i-1,i}V_r^T(B)W_{[2]}x_i(t)x_{i-1}(t))\Big]\\
&= \frac{1}{2}\Big[(e_{i,i+1}V_r^T(A) + e_{i+1,i}V_r^T(B)W_{[2]})D_f^{2^2,2^3}W_{[2^2,2^2]}\\
&\quad +(e_{i,i-1}V_r^T(A) + e_{i-1,i}V_r^T(B)W_{[2]})W_{[2]}D_f^{2^2,2^3}W_{[2,2^2]}\Big] \ltimes_{j=i-2}^{i+2} x_j(t)\\
&:= M_i \ltimes_{j=i-2}^{i+2} x_j(t),
\end{aligned}$$

where

$$M_i = \frac{1}{2}\Big[(e_{i,i+1}V_r^T(A) + e_{i+1,i}V_r^T(B)W_{[2]})D_f^{2^2,2^3}W_{[2^2,2^2]}$$

$$+(e_{i,i-1}V_r^T(A)+e_{i-1,i}V_r^T(B)W_{[2]})W_{[2]}D_f^{2^2,2^3}W_{[2,2^2]}\Big].$$

Similarly, we have

$$\begin{aligned} c_{i-1}(t) &:= M_{i-1}\ltimes_{j=i-2}^{i+2}x_j(t),\\ c_{i+1}(t) &:= M_{i+1}\ltimes_{j=i-2}^{i+2}x_j(t),\end{aligned}$$

where

$$\begin{aligned} M_{i-1} &= \frac{1}{2}\Big[(e_{i-1,i}V_r^T(A)+e_{i,i-1}V_r^T(B)W_{[2]})D_f^{2^2,2^3}W_{[2,2^2]}\\ &\quad +(e_{i-1,i-2}V_r^T(A)+e_{i-2,i-1}V_r^T(B)W_{[2]})W_{[2]}D_f^{2^2,2^3}\Big],\\ M_{i+1} &= \frac{1}{2}\Big[(e_{i+1,i+2}V_r^T(A)+e_{i+2,i+1}V_r^T(B)W_{[2]})D_r^{2^3,2^2}\\ &\quad +(e_{i+1,i}V_r^T(A)+e_{i,i+1}V_r^T(B)W_{[2]})W_{[2]}D_f^{2^2,2^3}W_{[2^2,2^2]}\Big].\end{aligned}$$

Obviously, $M_j\in\mathbb{R}_{1\times 2^5}$, $j=i-1,i,i+1$ is the structural matrix of average payoff c_j, $\mathrm{Col}_k(M_j), k\in[1:2^5]$ represents the benefit of player j under the profile $\delta_{2^5}^k$. Thus, comparing the average payoff of player i and his neighbors $i-1$ and $i+1$ is converted into comparing the corresponding columns of matrices M_{i-1}, M_i and M_{i+1}.

First, for sake of discussion, consider the profile $(x_{i-2},x_{i-1},x_i,x_{i+1},x_{i+2})=(1,1,1,2,1)$ as an example.

Combined with the above analysis, we have

$$\begin{aligned} \mathrm{Col}_3(M_{i-1}) &= \frac{1}{2}[(e_{i-1,i-2}+e_{i-1,i})a+(e_{i-2,i-1}+e_{i,i-1})b+2\alpha],\\ \mathrm{Col}_3(M_i) &= \frac{1}{2}[e_{i,i+1}m+e_{i+1,i}f+e_{i,i-1}a+e_{i-1,i}b+2\alpha],\\ \mathrm{Col}_3(M_{i+1}) &= \frac{1}{2}[(e_{i+1,i+2}+e_{i+1,i})e+(e_{i+2,i+1}+e_{i,i+1})s-2\beta].\end{aligned}$$

To make defector imitate the strategy of cooperator, we need that

$$\min\{\mathrm{Col}_3(M_{i-1}),\mathrm{Col}_3(M_i)\}>\max\{\mathrm{Col}_3(M_{i+1})\},$$

i.e. $\min\{\frac{1}{2}[2b+2\alpha],\ \frac{1}{2}[b+f+2\alpha]\}>\frac{1}{2}[2e-2\beta]$. Hence

$$\alpha+\beta>\frac{1}{2}[2e-b-f]. \tag{7.18}$$

Next, according to the strategy choices of players $i-1$, i and $i+1$, we divide the profiles of players $i-2$, $i-1$, i, $i+1$, $i+2$ in vector form into 8 groups :

$$\{\delta_{2^5}^1,\delta_{2^5}^2,\delta_{2^5}^{17},\delta_{2^5}^{18}\},\ \{\delta_{2^5}^3,\delta_{2^5}^4,\delta_{2^5}^{19},\delta_{2^5}^{20}\},\ \{\delta_{2^5}^5,\delta_{2^5}^6,\delta_{2^5}^{21},\delta_{2^5}^{22}\},\ \{\delta_{2^5}^7,\delta_{2^5}^8,\delta_{2^5}^{23},\delta_{2^5}^{24}\},$$

$$\{\delta_{2^5}^9,\delta_{2^5}^{10},\delta_{2^5}^{25},\delta_{2^5}^{26}\},\ \{\delta_{2^5}^{11},\delta_{2^5}^{12},\delta_{2^5}^{27},\delta_{2^5}^{28}\},\ \{\delta_{2^5}^{13},\delta_{2^5}^{14},\delta_{2^5}^{29},\delta_{2^5}^{30}\},\ \{\delta_{2^5}^{15},\delta_{2^5}^{16},\delta_{2^5}^{31},\delta_{2^5}^{32}\},$$

and at the same time, the profiles of players $i-1$, i and $i+1$ are respectively :

$$(1,1,1),(1,1,2),(1,2,1),(1,2,2),(2,1,1),(2,1,2),(2,2,1),(2,2,2).$$

Similar to the above discussion, when $x_{i-2}x_{i-1}x_ix_{i+1}x_{i+2}=\delta_{2^5}^j$, $j=4,19,20$, *i.e.* the second group except the profile $\delta_{2^5}^3$, we have the following inequalities:

$$\alpha+\beta > \frac{1}{2}[e-b-f], \tag{7.19}$$

$$\alpha+\beta > \frac{1}{2}[2e-b-f], \tag{7.20}$$

$$\alpha+\beta > \frac{1}{2}[e-b-f]. \tag{7.21}$$

In conclusion, when the strategy profile of players $i-1$, i and $i+1$ is $(1,1,2)$, to make defector imitates the strategy of cooperator, according to inequalities (7.18)-(7.21), we have

$$\alpha+\beta > \frac{1}{2}[2e-b-f]. \tag{7.22}$$

Similar results can be obtained for the third to seventh group, and the following inequalities are obtained:

$$\alpha+\beta > e-f, \tag{7.23}$$

$$\alpha+\beta > \frac{1}{2}[e-2f], \tag{7.24}$$

$$\alpha+\beta > \frac{1}{2}[2e-b-f], \tag{7.25}$$

$$\alpha+\beta > e-f, \tag{7.26}$$

$$\alpha+\beta > \frac{1}{2}[e-2f]. \tag{7.27}$$

Particularly, when both players $i-1$, i and $i+1$ select full cooperation (or full defection), *i.e.* for the first group (or the eighth group) strategy profiles, no matter what value of α and β, we all have $x_i(t+1)=\delta_2^1$ (or $x_i(t+1)=\delta_2^2$).

Based on the above analysis, for all strategy profiles, to make defector imitate the strategy of cooperator, according to inequalities (7.22)-(7.27), we need

$$\alpha+\beta > e-f,$$

i.e. when α, $\beta\in\{\alpha,\beta|\alpha+\beta>e-f\}$, then we have $|\Gamma_e|=2^n-1$.

7.5 An illustrative example

In this section, we provide an illustrative example to show the impact of reward and punishment parameters on the convergence domain of evolutionary boxed pig game with the mechanism of passive reward and punishment.

Example 7.5.1. *Consider a passive reward and punishment networked evolutionary boxed pig game with the following items :*

- *denote by* $N = \{1, 2, 3, 4, 5\}$ *the player set, and each player has the same strategy set* $S = \{1, 2\}$*, where* $1, 2$ *represent cooperation and defection respectively.*
- *the adjacency matrix is shown as follows :*

$$\begin{aligned} E &= (e_{ij})_{5\times 5} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}. \end{aligned}$$

- *the payoff bi-matrix is shown in Table 7.3, where* $e = 9, a = 5, m = s = 4, b = 1, f = -1, g = h = 0$*,* α *and* β *represent the amount of reward and punishment applied by the feeder to cooperator and defector, respectively, and* $\alpha \geq 0$*,* $\beta \geq 0$*.*

Table 7.3: Payoff bi-matrix of Example 7.5.1

$P_1 \setminus P_2$	$C = 1$	$D = 2$
$C = 1$	$(a + \alpha, b + \alpha)$	$(m + \alpha, s - \beta)$
$D = 2$	$(e - \beta, f + \alpha)$	$(g - \beta, h - \beta)$

Combined with the above analysis, to achieve the optimization goal, the sum of resources expended by reward and punishment should satisfy $\alpha + \beta > 10$. For example we can choose $\alpha = \beta = 6$. Then, according to the payoff bi-matrix, we have

$$\begin{aligned} V_r^T(A) &= [11 \ \ 10 \ \ 3 \ \ -6], \\ V_r^T(B) &= [7 \ \ -2 \ \ 5 \ \ -6]. \end{aligned}$$

In the following, we convert the dynamics of the game into an algebraic form.

First, the algebraic forms of average payoffs for all players are given by:

$$c_1(x(t)) \ = \ \frac{1}{2}\Big[V_r^T(B)W_{[2]}x_1(t)x_2(t) + V_r^T(B)W_{[2]}x_1(t)x_5(t)\Big] = P_1 x(t),$$

$$
\begin{aligned}
c_2(x(t)) &= \frac{1}{2}\Big[V_r^T(B)W_{[2]}x_2(t)x_3(t) + V_r^T(A)x_2(t)x_1(t)\Big] = P_2x(t),\\
c_3(x(t)) &= \frac{1}{2}\Big[V_r^T(B)W_{[2]}x_3(t)x_4(t) + V_r^T(A)x_3(t)x_2(t)\Big] = P_3x(t),\\
c_4(x(t)) &= \frac{1}{2}\Big[V_r^T(B)W_{[2]}x_4(t)x_5(t) + V_r^T(A)x_4(t)x_3(t)\Big] = P_4x(t),\\
c_5(x(t)) &= \frac{1}{2}\Big[V_r^T(A)x_5(t)x_1(t) + V_r^T(A)x_5(t)x_4(t)\Big] = P_5x(t),
\end{aligned}
$$

where $P_i \in \mathbb{R}_{1\times 2^5}$ is the structural matrix of $c_i(i \in [1:5])$, according to Algorithm 9, the structural matrix of average payoff for each player can be calculated as follows:

$$
\begin{aligned}
P_1 &= \frac{1}{2}\Big[V_r^T(B)W_{[2]}D_f^{2^2,2^3} + V_r^T(B)W_{[2]}W_{[2]}D_r^{2^3,2^2}W_{[2,2^4]}\Big]\\
&= [7\ \ 6\ \ 7\ \ 6\ \ 7\ \ 6\ \ 7\ \ 6\ \ 6\ \ 5\ \ 6\ \ 5\ \ 6\ \ 5\ \ 6\ \ 5\\
&\quad -2\ -4\ -2\ -4\ -2\ -4\ -2\ -4\ -4\ -6\ -4\ -6\ -4\ -6\ -4\ -6],\\
P_2 &= \frac{1}{2}\Big[V_r^T(B)W_{[2]}D_f^{2^2,2^3}W_{[2,2^4]} + V_r^T(A)W_{[2]}D_r^{2^3,2^2}W_{[2^2,2^3]}\Big]\\
&= [9\ \ 9\ \ 9\ \ 9\ \ 8\ \ 8\ \ 8\ \ 8\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ -1.5\ \ -1.5\ \ -1.5\ \ -1.5\\
&\quad 8.5\ \ 8.5\ \ 8.5\ \ 8.5\ \ 7.5\ \ 7.5\ \ 7.5\ \ 7.5\ -4\ -4\ -4\ -4\ -6\ -6\ -6\ -6],\\
P_3 &= \frac{1}{2}\Big[V_r^T(B)W_{[2]}D_f^{2^2,2^3}W_{[2^2,2^3]} + V_r^T(A)W_{[2]}D_r^{2^3,2^2}W_{[2^3,2^2]}\Big]\\
&= [9\ \ 9\ \ 8\ \ 8\ \ 0.5\ \ 0.5\ -1.5\ -1.5\ \ 8.5\ \ 8.5\ \ 7.5\ \ 7.5\ -4\ -4\ -6\ -6\\
&\quad 9\ \ 9\ \ 8\ \ 8\ \ 0.5\ \ 0.5\ -1.5\ -1.5\ \ 8.5\ \ 8.5\ \ 7.5\ \ 7.5\ -4\ -4\ -6\ -6],\\
P_4 &= \frac{1}{2}\Big[V_r^T(B)W_{[2]}D_f^{2^2,2^3}W_{[2^3,2^2]} + V_r^T(A)W_{[2]}D_r^{2^3,2^2}W_{[2^4,2]}\Big]\\
&= [9\ \ 7\ \ 0.5\ -1.5\ \ 8.5\ \ 7.5\ -4\ -6\ \ 9\ \ 7\ \ 0.5\ -1.5\ \ 8.5\ \ 7.5\ -4\ -6\\
&\quad 9\ \ 7\ \ 0.5\ -1.5\ \ 8.5\ \ 7.5\ -4\ -6\ \ 9\ \ 7\ \ 0.5\ -1.5\ \ 8.5\ \ 7.5\ -4\ -6],\\
P_5 &= \frac{1}{2}\Big[V_r^T(A)D_f^{2^2,2^3}W_{[2^4,2]} + V_r^T(A)W_{[2]}D_r^{2^3,2^2}\Big]\\
&= [11\ \ 3\ \ 10.5\ -1.5\ \ 11\ \ 3\ \ 10.5\ -1.5\ \ 11\ \ 3\ \ 10.5\ -1.5\ \ 11\ \ 3\ \ 10.5\ -1.5\\
&\quad 10.5\ -1.5\ \ 10\ -6\ \ 10.5\ -1.5\ \ 10\ -6\ \ 10.5\ -1.5\ \ 10\ -6\ \ 10.5\ -1.5\ \ 10\ -6].
\end{aligned}
$$

Calculate matrices H_i^k, F_i^k, $i \in [1:5]$, $k \in [1:2^5]$, respectively. Identify the column index ε_k^i. Then

$$
\begin{aligned}
L_1 &= \delta_2[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\\
&\qquad 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 1\ 2\ 1\ 2\ 2\ 2],\\
L_2 &= \delta_2[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\\
&\qquad 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2],\\
L_3 &= \delta_2[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\\
&\qquad 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2],
\end{aligned}
$$

$$\begin{aligned} L_4 &= \delta_2[1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2 \\ &\quad 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2], \\ L_5 &= \delta_2[1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2 \\ &\quad 1\ 1\ 1\ 2\ 1\ 1\ 1\ 2\ 1\ 1\ 1\ 2\ 1\ 1\ 1\ 2]. \end{aligned}$$

Therefore, the algebraic form of the game in this case is

$$x(t+1) = Lx(t), \tag{7.28}$$

where

$$\begin{aligned} L = \delta_{32}[&1\ 1\ 1\ 1\ 1\ 1\ 1\ 3\ 1\ 1\ 1\ 1\ 1\ 1\ 5\ 8 \\ &1\ 1\ 1\ 2\ 1\ 1\ 1\ 4\ 1\ 17\ 1\ 18\ 9\ 25\ 29\ 32] \end{aligned}$$

A simple calculation shows that there exist time $T = 3$, such that

$$\begin{aligned} L^T = \delta_{32}[&1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ &1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 32], \end{aligned}$$

and from Theorem 7.4.1, the convergence domain of the game is

$$\Gamma_e = \left\{\delta_{2^5}^i | \mathrm{Col}_i(L^3) = \delta_{2^5}^1, i \in [1 : 2^5]\right\}.$$

Hence, the number of elements in $\Gamma_e : |\Gamma_e| = 2^5 - 1$, *i.e.* all initial profiles (except full defection) are eventually stable to full cooperation profile.

7.6 Conclusions

In this chapter, we have investigated the model of networked evolutionary boxed pig game based on the mechanism of passive reward and punishment. Using the matrix expression of logic and STP, we have constructed the algebraic form of the evolutionary dynamics. Based on the algebraic form, we have obtained necessary and sufficient conditions under which the dynamic process of the game starting from any initial profile except full defection can converge to full cooperation profile, and analyzed the impact of reward and punishment parameters on the final cooperation level. The study of an illustrative example has shown that our main results in this chapter are effective.

Chapter 8

Conclusions and Future Research

8.1 Conclusions

Using STP method, this thesis investigates the stability and control of several classes of logical dynamic systems and the applications in game theory. Firstly, the stability of DKVLNs has been analyzed and the event-triggered control has been designed such that all the initial states can be globally stabilized to the target state. Secondly, the robust control invariance and robust set stabilization of MVLCNs have been studied. Thirdly, the finite-time robust set stability with probability 1 of PBNs and finite-time robust set stabilization with probability 1 of PBCNs have been investigated. Fourthly, the stabilization and set stabilization of SBCNs with periodic switching signals have been addressed. Finally, the theoretical results obtained were applied to investigate the optimization and global convergence of NEGs, and a number of novel results have been obtained. The detailed research results can be summarized as follows:

- The stability and event-triggered control of DKVLNs have been investigated. First, the algebraic formulations for DKVLNs and DKVLCNs under event-triggered control have been constructed respectively. Then, the criteria to detect the global stability have been presented based on the iteration of the evolutionary process. Furthermore, the antecedence solution technique has been introduced into the stability analysis and the necessary and sufficient conditions have been presented for the global stability of DKVLNs. Moreover, necessary and sufficient conditions for the global stabilization of DKVLCNs have been given, and a constructive procedure has been proposed to design the event-triggered state feedback control via the antecedence solution technique.

- The robust control invariance and robust set stabilization of MVLCNs have been studied. First, the dynamics of the disturbed MVLCNs have been converted into algebraic expression. Then, an algorithm has been presented to compute the LRCIS of any given set of MVLCNs, and all the possible state feedback controls have been determined to keep the robust control invariance. Moreover, necessary and sufficient conditions to detect the robust set stabilization of MVLCNs have been derived. Furthermore, using the antecedence solution technique, a constructive procedure has been provided to design all the time-optimal state feedback controls.

- The finite-time robust set stability with probability 1 of PBNs and finite-time robust set stabilization with probability 1 of PBCNs have been investigated. First, the algebraic forms of PBNs and PBCNs with disturbances have been given. Then, algorithms have been proposed to compute the LRIS with probability 1 and the LRCIS with probability 1. Moreover, necessary and sufficient conditions have been derived respectively to determine the finite-time robust set stability with probability 1 and finite-time robust set stabilization with probability 1 of PBCNs. Furthermore, a constructive algorithm has been presented to design all the time-optimal controllers.

- The stabilization and set stabilization of SBCNs with periodic switching signal have been studied. First, the algebraic formulation of periodic SBCNs has been constructed. Then, necessary and sufficient conditions have been presented to determine the solvability of the stabilization of periodic SBCNs, and the constructive procedures of open loop controller as well as the design algorithms of switching-signal-dependent state feedback controller via antecedence solution technique have been provided. Furthermore, the criteria have been given to detect the solvability of set stabilization of periodic SBCNs, and the design algorithm has been constructed to obtain the switching-signal-dependent state feedback set stabilizers.

- The event-triggered control design of NEGs with time-invariant delay in strategies has been investigated. First, using STP method, the evolutionary dynamics of NEGs with time-invariant delay in strategies have been converted into a DKVLCN with algebraic form. Then, necessary and sufficient conditions have been given to determine whether the evolutionary dynamics can globally converge to the only desired strategy profile. Moreover, all valid state feedback event-triggered controllers have been constructed to assure the global convergence of the desired only strategy profile of DNEGs. Furthermore, the number of all valid state feedback event-triggered controllers has been obtained.

- The algebraization and optimization of networked evolutionary boxed pig games with passive reward and punishment have been studied. First, the evolutionary dynamic process of this kind of games has been modeled as a BN, and an algorithm has been presented to construct the algebraic formulation of the game. Then, necessary and sufficient conditions have been presented to detect whether the final dynamic behavior of boxed pig game with the mechanism of passive reward and punishment can converge to full cooperation profile. Moreover, values of reward and punishment parameters have been obtained assuring that all the initial profiles except full defection can converge to full cooperation profile.

8.2 Future research

In this thesis, the stability analysis and control of several classes of logical dynamic systems and the applications in game theory were systematically investigated, and some new results were obtained. However, many theoretical and practical problems need to be further investigated in the future.

- The mathematical tool used in this thesis is STP. This method plays a very important role in the analysis and control design of finite-valued systems, based on which, any finite-valued system can be converted into the algebraic formulation. The dimension of the state transition matrix of the system increases exponentially with the number of network nodes, which is the greatest limitation of STP. Thus, how to effectively reduce the computational complexity of STP will be a very meaningful research topic.
- The time-invariant delay is considered for KVLCNs and NEGs. This kind of time delays may be too limited. In practice, the time-variant delays are more general. Thus, the impact of time-variant delays on the evolutionary dynamics of KVLCNs and NEGs is worthy of further study. Moreover, in order to reduce the control execution times, the event-triggered control scheme is chosen for DKVLCNs and DNEGs, whether there exists an adjustment method to further minimise the control times need to be further discussed.
- The results obtained in logical dynamic system without disturbances have been successfully applied to the analysis and control of NEGs. For the robust control theory of logical dynamic system, there are few researches on its application to NEGs. The investigation for NEGs with disturbances is meaningful. Some classical problems in NEGs are worthy of further study, such as the robust optimization problem, the robust stable degree analysis for profiles and the robust strategy consensus.

Appendix A

An Appendix

A.1 List of publications

- **Journal Papers**

1. **Jianjun Wang**, Renato De Leone, Shihua Fu, Jianwei Xia. Input-output decoupling for mix-valued logical control networks via the semi-tensor product method. *International Journal of Control*, 2021, 94(9), 2419-2427.

2. **Jianjun Wang**, Renato De Leone, Shihua Fu, Jianwei Xia, Lishan Qiao. Stabilisation and set stabilisation of periodic switched Boolean control networks. *International Journal of Control*, 2021, doi: 10.1080/00207179.2021.2009576.

3. **Jianjun Wang**, Renato De Leone, Shihua Fu, Jianwei Xia, Lishan Qiao. Event-triggered control design for networked evolutionary games with time invariant delay in strategies. *International Journal of Systems Science*, 2021, 52(3), 493-504.

4. **Jianjun Wang**, Wen Liu, Shihua Fu, Jianwei Xia. On robust set stability and set stabilization of probabilistic Boolean control networks. *Applied Mathematics and Computation*, 2022, 422, 126992.

5. **Jianjun Wang**, Jianli Zhao, Shihua Fu. Algebraization and optimization of networked evolutionary boxed pig games with passive reward and punishment. *Asian Journal of Control*, 2019, 21(5), 2415-2424.

6. **Jianjun Wang**, Shihua Fu, Renato De Leone, Jianwei Xia, Lishan Qiao. On robust control invariance and robust set stabilization of mix-valued logical control networks. *International Journal of Robust and Nonlinear Control*, minor revision.

- **Conference Papers**

7. **Jianjun Wang**, Renato De Leone, Shihua Fu, Jianwei Xia, Lishan Qiao. Input-output decoupling of singular Boolean control networks. *Proceedings of the 39th Chinese Control Conference*, Jul 27-29, 2020, Shengyang, China, 463-468.

A.2 List of projects

- National Natural Science Foundation of China, (Grant No.61976110), 2020.01.01-2023.12.31, Participant.
- Natural Science Foundation of Shandong Province, (Grant No.ZR2019BF023), 2019.07-2022.06, Participant.

References

Bibliography

[1] Tatsuya Akutsu, Satoru Miyano, and Satoru Kuhara. “Inferring qualitative relations in genetic networks and metabolic pathways”. In: *Bioinformatics* 16.8 (2000), pp. 727–734.

[2] Tatsuya Akutsu et al. “Control of Boolean networks: Hardness results and algorithms for tree structured networks”. In: *Journal of Theoretical Biology* 244.4 (2007), pp. 670–679.

[3] Réka Albert and Hans G Othmer. “The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster”. In: *Journal of Theoretical Biology* 223.1 (2003), pp. 1–18.

[4] Maximino Aldana. “Boolean dynamics of networks with scale-free topology”. In: *Physica D: Nonlinear Phenomena* 185.1 (2003), pp. 45–66.

[5] Guido Arnout and Hugo J De Man. “The use of threshold functions and Boolean-controlled network elements for macromodeling of LSI circuits”. In: *IEEE Journal of Solid-State Circuits* 13.3 (1978), pp. 326–332.

[6] Dario Bauso. “Boolean-controlled systems via receding horizon and linear programing”. In: *Mathematics of Control, Signals, and Systems* 21.1 (2009), pp. 69–91.

[7] Michael Bushnell and Vishwani Agrawal. *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits*. Vol. 17. Springer Science & Business Media, 2004.

[8] Bor-Sen Chen and Yu-Chao Wang. “On the attenuation and amplification of molecular noise in genetic regulatory networks”. In: *BMC Bioinformatics* 7.1 (2006), pp. 1–14.

[9] Dai-zhan Cheng, Yin Zhao, and Xiang-ru Xu. “Mix-valued logic and its applications”. In: *Journal of Shandong University (Natural Science)* 46.10 (2011), pp. 32–44.

[10] Daizhan Cheng. “Disturbance decoupling of Boolean control networks”. In: *IEEE Transactions on Automatic Control* 56.1 (2010), pp. 2–10.

[11] Daizhan Cheng. “Semi-tensor product of matrices and its application to Morgen’s problem”. In: *Science in China Series: Information Sciences* 44.3 (2001), pp. 195–212.

[12] Daizhan Cheng and Hongsheng Qi. "A linear representation of dynamics of Boolean networks". In: *IEEE Transactions on Automatic Control* 55.10 (2010), pp. 2251–2258.

[13] Daizhan Cheng and Hongsheng Qi. "Controllability and observability of Boolean control networks". In: *Automatica* 45.7 (2009), pp. 1659–1667.

[14] Daizhan Cheng, Hongsheng Qi, and Zhiqiang Li. *Analysis and control of Boolean networks: a semi-tensor product approach.* Springer Science & Business Media, 2010.

[15] Daizhan Cheng, Tingting Xu, and Hongsheng Qi. "Evolutionarily stable strategy of networked evolutionary games". In: *IEEE Transactions on Neural Networks and Learning Systems* 25.7 (2013), pp. 1335–1345.

[16] Daizhan Cheng, Yin Zhao, and Jiang-Bo Liu. "Optimal control of finite-valued networks". In: *Asian Journal of Control* 16.4 (2014), pp. 1179–1190.

[17] Daizhan Cheng et al. "Controllability of Boolean networks via mixed controls". In: *IEEE Control Systems Letters* 2.2 (2018), pp. 254–259.

[18] Daizhan Cheng et al. "Modeling, analysis and control of networked evolutionary games". In: *IEEE Transactions on Automatic Control* 60.9 (2015), pp. 2402–2415.

[19] Daizhan Cheng et al. "Stability and stabilization of Boolean networks". In: *International Journal of Robust and Nonlinear Control* 21.2 (2011), pp. 134–156.

[20] Gilles Didier, Elisabeth Remy, and Claudine Chaouiya. "Mapping multivalued onto Boolean dynamics". In: *Journal of Theoretical Biology* 270.1 (2011), pp. 177–184.

[21] Xueying Ding and Haitao Li. "Finite-time time-variant feedback stabilization of logical control networks with Markov jump disturbances". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 67.10 (2019), pp. 2079–2083.

[22] Xueying Ding et al. "Stochastic stability and stabilization of n-person random evolutionary Boolean games". In: *Applied Mathematics and Computation* 306 (2017), pp. 1–12.

[23] Glenn Ellison. "Learning, local interaction, and coordination". In: *Econometrica: Journal of the Econometric Society* (1993), pp. 1047–1071.

[24] Ettore Fornasini and Maria Elena Valcher. "Fault detection analysis of Boolean control networks". In: *IEEE Transactions on Automatic Control* 60.10 (2015), pp. 2734–2739.

[25] Ettore Fornasini and Maria Elena Valcher. "Observability, reconstructibility and state observers of Boolean control networks". In: *IEEE Transactions on Automatic Control* 58.6 (2012), pp. 1390–1401.

[26] Ettore Fornasini and Maria Elena Valcher. "On the periodic trajectories of Boolean control networks". In: *Automatica* 49.5 (2013), pp. 1506–1509.

[27] Ettore Fornasini and Maria Elena Valcher. "Optimal control of Boolean control networks". In: *IEEE Transactions on Automatic Control* 59.5 (2013), pp. 1258–1270.

[28] Daniel Friedman. "On economic applications of evolutionary game theory". In: *Journal of Evolutionary Economics* 8.1 (1998), pp. 15–43.

[29] Shihua Fu, Yuzhen Wang, and Guodong Zhao. "A matrix approach to the analysis and control of networked evolutionary games with bankruptcy mechanism". In: *Asian Journal of Control* 19.2 (2017), pp. 717–727.

[30] Shihua Fu, Jianli Zhao, and Jianjun Wang. "Input–output decoupling control design for switched Boolean control networks". In: *Journal of the Franklin Institute* 355.17 (2018), pp. 8576–8596.

[31] Peilian Guo and Changda Han. "Nash equilibrium and group strategy consensus of networked evolutionary game with coupled social groups". In: *Applied Mathematics and Computation* 409 (2021), p. 126380.

[32] Peilian Guo, Yuzhen Wang, and Haitao Li. "Algebraic formulation and strategy optimization for a class of evolutionary networked games via semi-tensor product method". In: *Automatica* 49.11 (2013), pp. 3384–3389.

[33] Peilian Guo, Yuzhen Wang, and Haitao Li. "Stable degree analysis for strategy profiles of evolutionary networked games". In: *Science China Information Sciences* 59.5 (2016), pp. 1–12.

[34] Peilian Guo et al. "Semi-tensor product method to a class of event-triggered control for finite evolutionary networked games". In: *IET Control Theory & Applications* 11.13 (2017), pp. 2140–2145.

[35] Yuqian Guo. "Observability of Boolean control networks using parallel extension and set reachability". In: *IEEE Transactions on Neural Networks and Learning Systems* 29.12 (2018), pp. 6402–6408.

[36] Yuqian Guo et al. "Set stability and set stabilization of Boolean control networks based on invariant subsets". In: *Automatica* 61 (2015), pp. 106–112.

[37] Yuqian Guo et al. "Stability and set stability in distribution of probabilistic Boolean networks". In: *IEEE Transactions on Automatic Control* 64.2 (2018), pp. 736–742.

[38] Ming Han, Yang Liu, and Yanshuai Tu. "Controllability of Boolean control networks with time delays both in states and inputs". In: *Neurocomputing* 129 (2014), pp. 467–475.

[39] Stephen E Harris et al. "A model of transcriptional regulatory networks based on biases in the observed regulation rules". In: *Complexity* 7.4 (2002), pp. 23–40.

[40] WPM Heemels Heemels, MCF Donkers, and Andrew R Teel. "Periodic event-triggered control for linear systems". In: *IEEE Transactions on Automatic Control* 58.4 (2012), pp. 847–861.

[41] Jack Heidel et al. "Finding cycles in synchronous Boolean networks with applications to biochemical systems". In: *International Journal of Bifurcation and Chaos* 13.03 (2003), pp. 535–552.

[42] Franziska Hinkelmann et al. "ADAM: analysis of discrete models of biological systems using computer algebra". In: *BMC Bioinformatics* 12.1 (2011), pp. 1–11.

[43] Sui Huang and Donald E Ingber. "Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks". In: *Experimental Cell Research* 261.1 (2000), pp. 91–103.

[44] Weini Huang et al. "Emergence of stable polymorphisms driven by evolutionary games between mutants". In: *Nature Communications* 3.1 (2012), pp. 1–7.

[45] Trey Ideker, Timothy Galitski, and Leroy Hood. "A new approach to decoding life: systems biology". In: *Annual Review of Genomics and Human Genetics* 2.1 (2001), pp. 343–372.

[46] Guangyu Jia, Min Meng, and Jun-e Feng. "Function perturbation of mix-valued logical networks with impacts on limit sets". In: *Neurocomputing* 207 (2016), pp. 428–436.

[47] Guangyu Jia et al. "Further Results for Pinning Stabilization of Boolean Networks". In: *IEEE Transactions on Control of Network Systems* (2020).

[48] Yingzhe Jia, Daizhan Cheng, and Jun-e Feng. "State feedback stabilization of generic logic systems via Ledley antecedence solution". In: *Mathematical Methods in the Applied Sciences* (2020).

[49] Stuart A Kauffman. "Metabolic stability and epigenesis in randomly constructed genetic nets". In: *Journal of Theoretical Biology* 22.3 (1969), pp. 437–467.

[50] Hiroaki Kitano. "Systems biology: a brief overview". In: *science* 295.5560 (2002), pp. 1662–1664.

[51] HK Lam et al. "Design of fuzzy observer controllers for multivariable uncertain systems using stability and robustness analysis". In: *Multi. Val. Logic* 5 (2000), pp. 391–405.

[52] Dmitriy Laschov and Michael Margaliot. "Minimum-time control of Boolean networks". In: *SIAM Journal on Control and Optimization* 51.4 (2013), pp. 2869–2892.

[53] Robert S Ledley. "Digital computational methods in symbolic logic, with examples in biochemistry". In: *Proceedings of the National Academy of Sciences of the United States of America* 41.7 (1955), p. 498.

[54] Benjamin Lewin. *Genes VII.* Oxford University Press, 1999.

[55] Bowen Li et al. "Event-triggered control for the disturbance decoupling problem of Boolean control networks". In: *IEEE Transactions on Cybernetics* 48.9 (2017), pp. 2764–2769.

[56] Bowen Li et al. "Robust invariant set analysis of Boolean networks". In: *Complexity* 2019 (2019).

[57] Fangfei Li and Jitao Sun. "Controllability of probabilistic Boolean control networks". In: *Automatica* 47.12 (2011), pp. 2765–2771.

[58] Fangfei Li and Jitao Sun. "Stability and stabilization of multivalued logical networks". In: *Nonlinear Analysis: Real World Applications* 12.6 (2011), pp. 3701–3712.

[59] Fangfei Li and Yang Tang. "Set stabilization for switched Boolean control networks". In: *Automatica* 78 (2017), pp. 223–230.

[60] Fangfei Li, Di Wang, and Yang Tang. "Pinning Controllability of k-Valued Logical Systems". In: *IEEE Transactions on Control of Network Systems* 7.3 (2020), pp. 1523–1533.

[61] Fangfei Li and Lihua Xie. "Set stabilization of probabilistic Boolean networks using pinning control". In: *IEEE Transactions on Neural Networks and Learning Systems* 30.8 (2018), pp. 2555–2561.

[62] Fangfei Li et al. "On stabilization and set stabilization of multivalued logical systems". In: *Automatica* 80 (2017), pp. 41–47.

[63] Haitao Li and Xueying Ding. "A control Lyapunov function approach to feedback stabilization of logical control networks". In: *SIAM Journal on Control and optimization* 57.2 (2019), pp. 810–831.

[64] Haitao Li, Pingping Song, and Qiqi Yang. "Pinning control design for robust output tracking of k-valued logical networks". In: *Journal of the Franklin Institute* 354.7 (2017), pp. 3039–3053.

[65] Haitao Li and Yuzhen Wang. "Minimum-time state feedback stabilization of constrained Boolean control networks". In: *Asian Journal of Control* 18.5 (2016), pp. 1688–1697.

[66] Haitao Li and Yuzhen Wang. "Output feedback stabilization control design for Boolean control networks". In: *Automatica* 49.12 (2013), pp. 3641–3645.

[67] Haitao Li and Yuzhen Wang. "Robust stability and stabilisation of Boolean networks with disturbance inputs". In: *International Journal of Systems Science* 48.4 (2017), pp. 750–756.

[68] Haitao Li, Yuzhen Wang, and Peilian Guo. "State feedback based output tracking control of probabilistic Boolean networks". In: *Information Sciences* 349 (2016), pp. 1–11.

[69] Haitao Li, Yuzhen Wang, and Zhenbin Liu. "A semi-tensor product approach to pseudo-Boolean functions with application to Boolean control networks". In: *Asian Journal of Control* 16.4 (2014), pp. 1073–1081.

[70] Haitao Li, Yuzhen Wang, and Lihua Xie. "Output tracking control of Boolean control networks via state feedback: Constant reference signal case". In: *Automatica* 59 (2015), pp. 54–59.

[71] Haitao Li, Lihua Xie, and Yuzhen Wang. "On robust control invariance of Boolean control networks". In: *Automatica* 68 (2016), pp. 392–396.

[72] Haitao Li, Lihua Xie, and Yuzhen Wang. "Output regulation of Boolean control networks". In: *IEEE Transactions on Automatic Control* 62.6 (2016), pp. 2993–2998.

[73] Haitao Li, Xiaojing Xu, and Xueying Ding. "Finite-time stability analysis of stochastic switched Boolean networks with impulsive effect". In: *Applied Mathematics and Computation* 347 (2019), pp. 557–565.

[74] Rui Li, Meng Yang, and Tianguang Chu. "State feedback stabilization for Boolean control networks". In: *IEEE Transactions on Automatic Control* 58.7 (2013), pp. 1853–1857.

[75] Rui Li, Meng Yang, and Tianguang Chu. "State feedback stabilization for probabilistic Boolean networks". In: *Automatica* 50.4 (2014), pp. 1272–1278.

[76] S. Li. "Analysis on the countermeasures of college talent incentive mechanism from the perspective of pigs' payoffs". In: *Value Engineering* 20 (2015), pp. 210–211.

[77] Yalu Li, Xueying Ding, and Haitao Li. "Robust consensus of networked evolutionary games with attackers and forbidden profiles". In: *Entropy* 20.1 (2018), p. 15.

[78] Yalu Li, Haitao Li, and Xueying Ding. "Set stability of switched delayed logical networks with application to finite-field consensus". In: *Automatica* 113 (2020), p. 108768.

[79] Yalu Li, Haitao Li, and Peiyong Duan. "Synchronization of switched logical control networks via event-triggered control". In: *Journal of the Franklin Institute* 355.12 (2018), pp. 5203–5216.

[80] Yalu Li, Haitao Li, and Yuanyuan Li. "Constrained set controllability of logical control networks with state constraints and its applications". In: *Applied Mathematics and Computation* 405 (2021), p. 126259.

[81] Yalu Li, Haitao Li, and Weiwei Sun. "Event-triggered control for robust set stabilization of logical control networks". In: *Automatica* 95 (2018), pp. 556–560.

[82] Yifeng Li and Jiandong Zhu. "On disturbance decoupling problem of Boolean control networks". In: *Asian Journal of Control* 21.6 (2019), pp. 2543–2550.

[83] Lin Lin, Jinde Cao, and Leszek Rutkowski. "Robust event-triggered control invariance of probabilistic Boolean control networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 31.3 (2019), pp. 1060–1065.

[84] Huachao Liu et al. "Observability of Boolean networks via STP and graph methods". In: *IET Control Theory & Applications* 13.7 (2019), pp. 1031–1037.

[85] Rongjian Liu et al. "Delayed feedback control for stabilization of Boolean control networks with state delay". In: *IEEE Transactions on Neural Networks and Learning Systems* 29.7 (2017), pp. 3283–3288.

[86] Rongjian Liu et al. "Output feedback control for set stabilization of Boolean control networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 31.6 (2019), pp. 2129–2139.

[87] Yang Liu et al. "Sampled-data state feedback stabilization of Boolean control networks". In: *Neural Computation* 28.4 (2016), pp. 778–799.

[88] Zhenbin Liu and Yuzhen Wang. "Disturbance decoupling of mix-valued logical networks via the semi-tensor product method". In: *Automatica* 48.8 (2012), pp. 1839–1844.

[89] Zhenbin Liu, Yuzhen Wang, and Daizhan Cheng. "Nonsingularity of feedback shift registers". In: *Automatica* 55 (2015), pp. 247–253.

[90] Zhenbin Liu, Yuzhen Wang, and Haitao Li. "New approach to derivative calculation of multi-valued logical functions with application to fault detection of digital circuits". In: *IET Control Theory & Applications* 8.8 (2014), pp. 554–560.

[91] Jianquan Lu et al. "Event-triggered sampled feedback synchronization in an array of output-coupled Boolean control networks". In: *IEEE Transactions on Cybernetics* (2019).

[92] Jianquan Lu et al. "On pinning controllability of Boolean control networks". In: *IEEE Transactions on Automatic Control* 61.6 (2015), pp. 1658–1663.

[93] Jianquan Lu et al. "Pinning stabilization of Boolean control networks via a minimum number of controllers". In: *IEEE Transactions on Cybernetics* (2019).

[94] Jianquan Lu et al. "Stabilization of Boolean control networks under aperiodic sampled-data control". In: *SIAM Journal on Control and Optimization* 56.6 (2018), pp. 4385–4404.

[95] Jianquan Lu et al. "Synchronization analysis of master-slave probabilistic Boolean networks". In: *Scientific reports* 5.1 (2015), pp. 1–18.

[96] C. Xu M. Ji. "Cooperative behavior in evolutionary snowdrift game". In: *Journal of Suzhou University* 26.1 (2010), pp. 57–65.

[97] Jin Ma et al. "Approximation of the boundary of power system stability region based on semi-tensor theory, Part One: theoretical basis". In: *Automation of Electric Power Systems* 30.10 (2006), pp. 1–5.

[98] Ying Mao et al. "Stabilization of evolutionary networked games with length-r information". In: *Applied Mathematics and Computation* 337 (2018), pp. 442–451.

[99] Min Meng, Lu Liu, and Gang Feng. "Stability and l_1 gain analysis of Boolean networks with Markovian jump parameters". In: *IEEE Transactions on Automatic Control* 62.8 (2017), pp. 4222–4228.

[100] Min Meng et al. "Stability and stabilization of Boolean networks with stochastic delays". In: *IEEE Transactions on Automatic Control* 64.2 (2018), pp. 790–796.

[101] John F Nash et al. "Equilibrium points in n-person games". In: *Proceedings of the national academy of sciences* 36.1 (1950), pp. 48–49.

[102] Martin A Nowak and Karl Sigmund. "Evolutionary dynamics of biological games". In: *Science* 303.5659 (2004), pp. 793–799.

[103] Paul Nurse. "A long twentieth century of the cell cycle and beyond". In: *Cell* 100.1 (2000), pp. 71–78.

[104] Hisashi Ohtsuki et al. "A simple rule for the evolution of cooperation on graphs and social networks". In: *Nature* 441.7092 (2006), pp. 502–505.

[105] Jinfeng Pan et al. "Input-output decoupling of Boolean control networks". In: *Asian Journal of Control* 20.6 (2018), pp. 2185–2194.

[106] Hongsheng Qi and Yupeng Qiao. "Dynamics and control of singular boolean networks". In: *Asian Journal of Control* 21.6 (2019), pp. 2604–2613.

[107] Yupeng Qiao, Hongsheng Qi, and Daizhan Cheng. "Partition-based solutions of static logical networks with applications". In: *IEEE Transactions on Neural Networks and Learning Systems* 29.4 (2017), pp. 1252–1262.

[108] Eric Rasmusen. *Games and information: An introduction to game theory.* 1989.

[109] Fengli Ren and Jinde Cao. "Asymptotic and robust stability of genetic regulatory networks with time-varying delays". In: *Neurocomputing* 71.4-6 (2008), pp. 834–842.

[110] Ilya Shmulevich et al. "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks". In: *Bioinformatics* 18.2 (2002), pp. 261–274.

[111] Tao Sun et al. "Partial and global stabilization at an attractor for k-valued logical control networks". In: *Journal of the Franklin Institute* 357.11 (2020), pp. 7003–7019.

[112] Wei Sun et al. "Research on attack and deference in information security based on evolutionary game". In: *Information Science* 26 (2008), pp. 1408–1412.

[113] Hui Tian et al. "Stabilization of k-valued logical control networks by open-loop control via the reverse-transfer method". In: *Automatica* 83 (2017), pp. 387–390.

[114] RM Tong, MB Beck, and A Latten. "Fuzzy control of the activated sludge wastewater treatment process". In: *Automatica* 16.6 (1980), pp. 695–701.

[115] John J Tyson, Attila Csikasz-Nagy, and Bela Novak. "The dynamics of cell cycle regulation". In: *Bioessays* 24.12 (2002), pp. 1095–1109.

[116] Alan Veliz-Cuba and Brandilyn Stigler. "Boolean models can explain bistability in the lac operon". In: *Journal of Computational Biology* 18.6 (2011), pp. 783–794.

[117] Mitchell M Waldrop. *Complexity: The emerging science at the edge of order and chaos*. Simon and Schuster, 1993.

[118] Jianjun Wang et al. "Input–output decoupling for mix-valued logical control networks via the semi-tensor product method". In: *International Journal of Control* 94.9 (2021), pp. 2419–2427.

[119] Shuling Wang and Haitao Li. "New Results on the Disturbance Decoupling of Boolean Control Networks". In: *IEEE Control Systems Letters* 5.4 (2020), pp. 1157–1162.

[120] Shuling Wang et al. "Event-triggered control for disturbance decoupling problem of mix-valued logical networks". In: *Journal of the Franklin Institute* 357.2 (2020), pp. 796–809.

[121] Yuanhua Wang and Daizhan Cheng. "Dynamics and stability for a class of evolutionary games with time delays in strategies". In: *Science China Information Sciences* 59.9 (2016), pp. 1–10.

[122] Yuanhua Wang and Peilian Guo. "Optimal control of singular Boolean control networks via Ledley solution method". In: *Journal of the Franklin Institute* (2021).

[123] Eyal Weiss, Michael Margaliot, and Guy Even. "Minimal controllability of conjunctive Boolean networks is NP-complete". In: *Automatica* 92 (2018), pp. 56–62.

[124] Yuhu Wu, Madan Kumar, and Tielong Shen. "A stochastic logical system approach to model and optimal control of cyclic variation of residual gas fraction in combustion engines". In: *Applied Thermal Engineering* 93 (2016), pp. 251–259.

[125] Yuhu Wu et al. "Optimal control of Boolean control networks with average cost: A policy iteration approach". In: *Automatica* 100 (2019), pp. 378–387.

[126] Xiangru Xu and Yiguang Hong. "Matrix approach to model matching of asynchronous sequential machines". In: *IEEE Transactions on Automatic Control* 58.11 (2013), pp. 2974–2979.

[127] Xiaojing Xu et al. "Robust set stabilization of Boolean control networks with impulsive effects". In: *Nonlinear Analysis: Modelling and Control* 23.4 (2018), pp. 553–567.

[128] Jiaojiao Yang et al. “Synchronization of drive-response Boolean control networks with impulsive disturbances”. In: *Applied Mathematics and Computation* 364 (2020), p. 124679.

[129] Meng Yang, Rui Li, and Tianguang Chu. “Controller design for disturbance decoupling of Boolean control networks”. In: *Automatica* 49.1 (2013), pp. 273–277.

[130] Amol Yerudkar, Carmen Del Vecchio, and Luigi Glielmo. “Feedback stabilization control design for switched Boolean control networks”. In: *Automatica* 116 (2020), p. 108934.

[131] Amol Yerudkar, Carmen Del Vecchio, and Luigi Glielmo. “Output tracking control design of switched Boolean control networks”. In: *IEEE Control Systems Letters* 4.2 (2019), pp. 355–360.

[132] Yongyuan Yu, Min Meng, and Jun-e Feng. “Observability of Boolean networks via matrix equations”. In: *Automatica* 111 (2020), p. 108621.

[133] Jing Zhang et al. “Dynamics and convergence of hyper-networked evolutionary games with time delay in strategies”. In: *Information Sciences* 563 (2021), pp. 166–182.

[134] Kuize Zhang and Karl Henrik Johansson. “Efficient verification of observability and reconstructibility for large Boolean control networks with special structures”. In: *IEEE Transactions on Automatic Control* 65.12 (2020), pp. 5144–5158.

[135] Lequn Zhang et al. “Further results on disturbance decoupling of mix-valued logical networks”. In: *IEEE Transactions on Automatic Control* 59.6 (2013), pp. 1630–1634.

[136] Lijun Zhang and Kuize Zhang. “Controllability and observability of Boolean control networks with time-variant delays in states”. In: *IEEE Transactions on Neural Networks and Learning Systems* 24.9 (2013), pp. 1478–1484.

[137] Qiliang Zhang et al. “Event-triggered mechanism of designing set stabilization state feedback controller for switched Boolean networks”. In: *Applied Mathematics and Computation* 383 (2020), p. 125372.

[138] Xiao Zhang, Yuanhua Wang, and Daizhan Cheng. “Output tracking of Boolean control networks”. In: *IEEE Transactions on Automatic Control* 65.6 (2019), pp. 2730–2735.

[139] Guodong Zhao and Yuzhen Wang. “Formulation and optimization control of a class of networked evolutionary games with switched topologies”. In: *Nonlinear Analysis: Hybrid Systems* 22 (2016), pp. 98–107.

[140] Guodong Zhao et al. “Modelling and strategy consensus for a class of networked evolutionary games”. In: *International Journal of Systems Science* 49.12 (2018), pp. 2548–2557.

[141] Guodong Zhao et al. “Survey on applications of semi-tensor product method in networked evolutionary games”. In: *Journal of Applied Analysis & Computation* 10.1 (2020), pp. 32–54.

[142] Yin Zhao and DaiZhan Cheng. “On controllability and stabilizability of probabilistic Boolean control networks”. In: *Science China Information Sciences* 57.1 (2014), pp. 1–14.

[143] Yin Zhao, Zhiqiang Li, and Daizhan Cheng. “Optimal control of logical control networks”. In: *IEEE Transactions on Automatic Control* 56.8 (2010), pp. 1766–1776.

[144] Yin Zhao, Hongsheng Qi, and Daizhan Cheng. “Input-state incidence matrix of Boolean control networks and its applications”. In: *Systems & Control Letters* 59.12 (2010), pp. 767–774.

[145] Jie Zhong et al. “On the ensemble controllability of Boolean control networks using STP method”. In: *Applied Mathematics and Computation* 358 (2019), pp. 51–62.

[146] Jie Zhong et al. “Synchronization in an array of output-coupled Boolean networks with time delay”. In: *IEEE Transactions on Neural Networks and Learning Systems* 25.12 (2014), pp. 2288–2294.

[147] Rongpei Zhou et al. “Asymptotical feedback set stabilization of probabilistic Boolean control networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 31.11 (2019), pp. 4524–4537.

[148] Bing Zhu, Xiaohua Xia, and Zhou Wu. “Evolutionary game theoretic demand-side management and control for a class of networked smart grid”. In: *Automatica* 70 (2016), pp. 94–100.

[149] Qunxi Zhu et al. “Controllability and observability of boolean control networks via sampled-data control”. In: *IEEE Transactions on Control of Network Systems* 6.4 (2018), pp. 1291–1301.

[150] Qunxi Zhu et al. “Further results on the controllability of Boolean control networks”. In: *IEEE Transactions on Automatic Control* 64.1 (2018), pp. 440–442.

[151] Qunxi Zhu et al. “Observability of Boolean control networks”. In: *Science China Information Sciences* 61.9 (2018), pp. 1–12.

[152] Qunxi Zhu et al. “On the optimal control of Boolean control networks”. In: *SIAM Journal on Control and Optimization* 56.2 (2018), pp. 1321–1341.

[153] Shiyong Zhu et al. “Sampled-data state feedback control for the set stabilization of Boolean control networks”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50.4 (2018), pp. 1580–1589.

[154] Shiyong Zhu et al. “Stabilization of logical control networks: An event-triggered control approach”. In: *Science China Information Sciences* 63.1 (2020), pp. 1–11.

[155] Martín G Zimmermann and Víctor M Eguíluz. “Cooperation, social networks, and the emergence of leadership in a prisoner’s dilemma with adaptive local interactions”. In: *Physical Review E* 72.5 (2005), p. 056118.

[156] Yunlei Zou and Jiandong Zhu. “Cycles of periodically time-variant Boolean networks”. In: *Automatica* 51 (2015), pp. 175–179.

[157] Yunlei Zou and Jiandong Zhu. “Kalman decomposition for Boolean control networks”. In: *Automatica* 54 (2015), pp. 65–71.