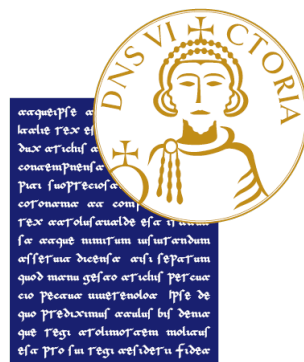


UNIVERSITY OF SANNIO
DEPARTMENT OF ENGINEERING

Ph.D. in
Information Technology for Engineering



Identification and Control of Two Classes of
Boolean Networks and the Applications in Game
Theory

SUPERVISORS:

Prof. **Carmen Del Vecchio** *Carmen Del Vecchio*
Prof. **Yuhu Wu** (Dalian University of Technology) *Yuhu Wu*

PH.D. COORDINATOR:

Prof. **Massimiliano Di Penta**

Massimiliano Di Penta

PH.D. CANDIDATE:

Chunfeng Jiang
Matr:D60030008

Jiang Chunfeng

PH.D. CYCLE: XXXVIII
ACADEMIC YEAR 2025/2026

Contents

List of Symbols	ii
List of Acronyms	iv
Abstract	vi
Acknowledgments	viii
1 Introduction	1
1.1 Background and Motivations	1
1.2 Literature Review	3
1.2.1 Several Fundamental Properties of BNs and SBNs	3
1.2.2 Output Tracking Control of BNs and SBNs	6
1.2.3 Optimization Control of NEGs	7
1.3 From Critical Gaps to Thesis Contributions	9
1.4 Main Contents and Thesis Organization	10
2 Preliminaries	13
2.1 Semi-tensor Product of Matrices	13
2.1.1 Definition and Basic Properties	13
2.1.2 Algebraic Representation of Logical Functions	14
2.2 Algebraic Representation of BNs and BCNs	15
2.3 Algebraic Representation of SBNs and SBCNs	17
2.4 NEGs and Hypergraphs	18
3 Identification of Switched Boolean Networks	21
3.1 Problem Formulation	22
3.2 Reachability and Observability of SBNs	24
3.3 Identification of SBNs under a Single-sample Scenario	31
3.4 Identification of SBNs under a Multiple-sample Scenario	35
3.5 Examples	38
3.6 Conclusion	43

4	Recurrent Output Tracking of Boolean Networks	45
4.1	Introduction	45
4.2	Problem Formulation	46
4.3	Recurrent Output Tracking of BNs	47
4.4	Recurrent Output Tracking of BCNs	50
4.5	Recurrent Periods of BNs and BCNs	56
4.6	Examples	58
4.7	Conclusion	63
5	Recurrent Output Tracking of Switched Boolean Control Networks	65
5.1	Introduction	65
5.2	Problem Formulation	66
5.3	Recurrent Output Tracking of SBCNs	67
5.4	An Example	73
5.5	Conclusion	76
6	Strategy Optimization of Controlled Heterogeneous Hyper-networked Evolutionary Games	77
6.1	Introduction	77
6.2	Algebraic Formulation of Heterogeneous HNEGs	78
6.3	Strategy Optimization	84
6.4	An Example	96
6.5	Conclusion	99
7	Conclusion and Future Work	101
7.1	Conclusion	101
7.2	Future Work	102
A	List of Publications	105
A.1	Author's Publication	105

List of Symbols

$\mathbb{N}$	The set of natural numbers
$\mathbb{Z}_+$	The set of positive integers
$\mathbb{R}^n$	n -dimensional Euclidean space
$\mathbb{R}_{m \times n}$	The set of $m \times n$ real matrices
$\mathcal{B}_{m \times n}$	The set of $m \times n$ Boolean matrices
$\mathcal{D}$	$\{0, 1\}$
$\mathcal{D}_k$	$\{1, 2, \dots, k\}$
$\mathcal{D}^n$	$\underbrace{\mathcal{D} \times \dots \times \mathcal{D}}_n$
$(A)_{i,j}$	The (i, j) -th element of matrix A
$\text{Col}_i(A)$	The i -th column of matrix A
$\text{Col}(A)$	The set of columns of matrix A
$\text{Row}_i(A)$	The i -th row of matrix A
$\text{Blk}_i(A)$	The i -th block of matrix A
δ_n^k	The k -th column of the identity matrix I_n
Δ_n	The set of logical columns $\{\delta_n^k \mid k \in \{1, 2, \dots, n\}\}$
Δ	Δ_2
$\delta_m[i_1, \dots, i_n]$	The logical matrix $[\delta_m^{i_1}, \dots, \delta_m^{i_n}]$
$\delta_m\{i_1, \dots, i_n\}$	The sequence $\{\delta_m^{i_1}, \dots, \delta_m^{i_n}\}$
$\mathcal{L}_{m \times n}$	The set of $m \times n$ logical matrices $\{A \mid \text{Col}(A) \subseteq \Delta_m\}$
$ Q $	The cardinal number of set Q
A^T	The transpose of matrix A
$A^{(m)}$	$\underbrace{A \times \dots \times A}_m$
$\mathbf{c}_m$	A column vector $\underbrace{[c, \dots, c]}_m^T$ where $c \in \{0, 1\}$
$\mathbf{1}_{m \times n}$	$\underbrace{[\mathbf{1}_m, \mathbf{1}_m, \dots, \mathbf{1}_m]}_n$
$\mathbf{0}_{m \times n}$	$\underbrace{[\mathbf{0}_m, \mathbf{0}_m, \dots, \mathbf{0}_m]}_n$
$\ltimes$	The semi-tensor product of matrices
$\otimes$	The Kronecker product of matrices
$*$	The Khatri-Rao product of matrices
$\circ$	The Hadamard product of matrices
$\text{lcm}(a, b)$	The least common multiple of a and b

$\lceil a/b \rceil$	The ceiling of a divided by b
Σ	The matrix summation
$+_{\mathcal{B}}$	Boolean addition of matrices (using logical OR $\vee$)
$\times_{\mathcal{B}}$	Boolean product of matrices (using logical AND $\wedge$)
$\Sigma_{\mathcal{B}}$	The Boolean summation
$A^{(n)}$	The n -th power of matrix A
$A \leq B$	Component-wise inequality $(A)_{i,j} \leq (B)_{i,j}$
$M _{\Lambda} \leq N _{\Lambda}$	Column-wise inequality restricted to the index set Λ
$Y \succ \mathbf{0}_m$	$(Y)_i \geq 0$ for all i and $\exists j, (Y)_j > 0$

List of Acronyms

BN	Boolean Network
BCN	Boolean Control Network
SBN	Switched Boolean Network
SBCN	Switched Boolean Control Network
STP	Semi-Tensor Product of Matrices
OT	Output Tracking
OTP	Output Tracking Problem
ROT	Recurrent Output Tracking
ROTP	Recurrent Output Tracking Problem
NEG	Networked Evolutionary Game
HNEG	Hyper-networked Evolutionary Game

Abstract

Boolean networks (BNs), also referred to as two-valued logical dynamical systems, provide a powerful discrete dynamical framework for modeling and analyzing complex systems in biology, engineering, and computer science. They also serve as an effective representation for bi-strategy evolutionary dynamics in finite-player games. Evolutionary game theory on complex networks has emerged as an important framework for studying collective behavior in systems with strategic interactions. In particular, the introduction of higher-order interactions via heterogeneous hypergraphs significantly enhances the modeling capability of evolutionary games. Evolutionary games on heterogeneous hypergraphs can be formulated as multi-valued logical dynamical systems. By resorting to the semi-tensor product (STP) of matrices, logical dynamical systems can be equivalently transformed into discrete-time linear algebraic systems. This algebraic representation also provides a natural bridge between BNs and evolutionary game dynamics. Based on that, this thesis develops an algebraic framework for the identification and control of BNs and switched Boolean networks (SBNs), and further extends it to heterogeneous hyper-networked evolutionary games (HNEGs). The main contributions are summarized as follows:

- The identification problem for SBNs is investigated under general switching structures, where switching signals affect both the state evolution and the corresponding output behavior. The reachability and observability properties of SBNs are first analyzed, and corresponding conditions are established. Based on these properties, criteria for SBN identification under both single-sample and multi-sample scenarios are derived. By incorporating temporal information, the exact correspondence between system states and outputs is established, and efficient algorithms are developed for the identification procedure. Interestingly, an SBN may remain identifiable even when some of its subnetworks are unidentifiable.
- A novel control objective, termed recurrent output tracking (ROT), is intro-

duced for BNs and Boolean control networks (BCNs). This formulation relaxes conventional tracking requirements by allowing system outputs to intermittently and infinitely track a desired reference signal. Algebraic solvability conditions are derived, and constructive state feedback controllers are designed. Moreover, the periodicity properties of BNs and BCNs are analyzed, showing that the classical tracking problem is a special case of the recurrent output tracking problem (ROTP).

- The ROTP is extended to switched Boolean control networks (SBCNs). By introducing the concepts of recurrent states and reachable sets, necessary and sufficient conditions are established for the solvability of the ROTP for SBCNs. An efficient algorithm is further developed to design state feedback controllers and switching signals such that the system can recurrently track a desired reference output. This highlights the fundamental role of switching in determining system reachability and ROT performance.
- Algebraic expression of heterogeneous HNEGs is established via the STP method, which overcomes the dimensional inconsistency of payoff functions and captures genuine multi-player interactions induced by hyperedges. By adding control players to the game, a controlled heterogeneous HNEG is introduced, and the strategy optimization problem is reformulated as a minimum-time set controllability problem. Necessary and sufficient conditions are derived to guarantee that all players' payoffs simultaneously reach and remain above the prescribed threshold. Furthermore, a state feedback controller is designed to achieve the objective in minimum time.

Acknowledgments

First and foremost, I would like to express my deepest gratitude and appreciation to my supervisor, Prof. Carmen Del Vecchio, for her patient guidance, constant encouragement, and the generous sharing of her rich expertise throughout my doctoral research. Particularly, she is always kind and very concerned about my life in Italy. I could not imagine a better supervisor for my doctoral study. Without her consistent support, this thesis would never have been accomplished.

I am grateful to my co-supervisor, Prof. Yuhu Wu, for giving me the opportunity to conduct research abroad. He took precious time to meet with me and gave me a lot of insightful guidance. His rigorous academic style set a benchmark for my progress.

Importantly, a special thanks to Prof. June Feng, Prof. Shihua Fu and Dr. Biao Wang 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.

I am also very lucky to have had the support of all my friends and colleagues from the Group for Research on Automatic Control (GRACE) laboratory. I have benefited a lot from the pleasant academic atmosphere and frequent academic exchanges. Their companionship and assistance have made my doctoral life fulfilling and memorable.

I am also very thankful to Enza Curcio, Muhammad Usman, and Babar Ali, for sharing endless advice on study and life in Benevento. I would also like to thank my Chinese friends in Benevento. In particular, many thanks to Siyu Zou, Ying Zhai, Zihao Dai and Jiawei Chen. With their openness and warmth, the time I spent in Benevento was enjoyable and happy.

Finally, I owe my deepest gratitude to my family for their unconditional love, understanding, and endless support over the years. Their encouragement has always been the strongest motivation for me to pursue my academic dream.

Chapter 1

Introduction

1.1 Background and Motivations

Boolean networks (BNs) were originally introduced by Kauffman in 1969 to model and analyze gene regulation processes [1]. Genes in a cell can be viewed as state nodes of the BN, and the regulatory relationships between genes can be characterized by the edges of this network. BNs are discrete-time dynamical systems whose states are represented as binary variables (0 and 1). The interaction between the various state nodes can be simulated using Boolean functions, and the value of each state node at time $t+1$ can be obtained from the value of a neighboring state node at time t through a series of complex logical operations.

To better guide disease treatment and drug preparation, the Boolean control networks (BCNs) emerged [2], where control inputs are introduced into BNs to influence the state evolution. Except in biological systems, both BNs and BCNs have attracted a lot of attention in digital communication and computing systems, social or economic systems, robotics, game theory and so on [3, 4, 5].

The switching phenomena is common in biological systems, for instance, eukaryotic cell division and growth are typically represented by a sequence comprising four distinct processes [6]. Similarly, gene expression or substances concentration, as in the *Drosophila* segmentation polarity gene network, are often modeled as switching signals [7]. Such complex phenomena cannot be adequately captured by a single BN model. To better simulate such dynamics, switched Boolean networks (SBNs) have been proposed as a natural extension of BNs, where the system evolution is governed by a family of subnetworks and a switching signal that determines the active dynamics at each time instant. By adding control inputs to an SBN, it becomes a switched Boolean control network (SBCN). To date, notable progress has

been made in understanding SBNs and SBCNs [8, 9, 10]. It is worth noting that a switching signal can be associated with different properties or states of a system, potentially depending on time, state conditions, or external inputs. While switching signals typically affect states, in more general contexts, they may also impact outputs [11, 12]. The presence of switching significantly enhances modeling capability, but also brings substantial challenges in system identification and control.

Many complex systems are inherently multi-player and involve strategic interactions among individuals. Evolutionary game theory provides an effective framework for modeling such systems [13], and networked evolutionary games (NEGs), first introduced by Nowak and May [14], further incorporate interaction structures into the dynamics. In an NEG, the interaction topology is explicitly encoded by a graph, where players are represented as nodes and their pairwise interactions as edges. Each player updates its strategy based on its neighbors' information. BN models can also be used to describe the evolutionary dynamics of NEGs where each player has two strategies. Furthermore, in finite NEGs, if the set of strategies a player can choose is finite, and the evolution of the game depends on limited historical information from the previous time step, then the evolution dynamics can be described by a logical dynamical system. A logical dynamical system refers to a system where the independent variables can only take a finite number of values, including binary classical two-valued logical networks (i.e., BNs), and more general multi-valued logical networks. This demonstrates the natural connection between BNs and NEGs. In recent years, with the development of complex network theory, NEGs have found extensive applications in diverse fields such as social cooperation and energy demand response [15, 16].

However, most existing NEG models are based on pairwise interactions, which are insufficient to describe higher-order interactions arising in applications such as group cooperation and multi-player decision-making [17, 18, 19, 20]. To address this limitation, hypergraphs have been introduced to model multi-player interactions [21], leading to hyper-networked evolutionary games (HNEGs) [22]. It is noted that a hyper-network is referred to as a network endowed with a hypergraph topological structure [23]. In particular, heterogeneous hypergraphs, which capture variations in both interaction size and connectivity (i.e., degree and order heterogeneity), provide a more realistic description of complex systems [24, 25], but significantly increase the analytical and control complexity.

BNs and HNEGs are both logical evolutionary processes described based on logical judgment, reasoning, and decision-making. Although logical expressions can

reveal the evolutionary process, purely logical approaches often suffer from limited scalability and lack of systematic design methodologies. For example, the results regarding fixed points and cycles in BNs are only applicable to specific systems [26, 27], and research on NEG's mainly utilizes median methods [28] and computer simulations [29], so that it is difficult to analyze the dynamic behavior of each player. The introduction of the semi-tensor product (STP) of matrices has significantly advanced the study of logical systems by enabling the transformation of logical dynamics into equivalent algebraic representations [30]. This mathematical tool was first proposed by Cheng to deal with Moore's problem in 2001 [31]. It allows logical dynamics to be analyzed using matrix-based tools [32], thereby facilitating the investigation of fundamental properties and control problems such as controllability and observability [33, 34], stability and stabilization [35, 36, 37], decoupling [8, 38], synchronization [39, 40], and optimal control [41, 42] within a unified setting. In addition, a comprehensive survey on the engineering applications of the STP method can be found in [43].

Motivated by the above considerations, this thesis investigates the identification and control of BNs and SBNs, and the applications to evolutionary games with higher-order interaction. The main focus is placed on three interrelated problems: the identification of SBNs, the recurrent output tracking (ROT) of BNs, BCNs and SBCNs, and the strategy optimization of heterogeneous HNEG's. These problems are interconnected through the common algebraic framework provided by the STP approach, which enables a unified treatment of logical dynamics and evolutionary game processes.

1.2 Literature Review

Within the extensive literature on BN and NEG models, we can identify three main research areas particularly relevant to our discussion, including some fundamental properties and tracking problems of BNs and SBNs, and optimization problems of NEG's. In this section, we will address each of these areas, briefly summarizing the key results from relevant research and the literature.

1.2.1 Several Fundamental Properties of BNs and SBNs

Identification plays a fundamental role in understanding and controlling complex systems, as it seeks to infer a system's structure, dynamics, or behavioral characteristics through the analysis and processing of specific data or information. Extensive

research has been conducted on identifying different kinds of systems, including linear systems and nonlinear systems [44, 45, 46]. As an important subclass of nonlinear systems, BNs have also garnered significant attention in this area [47, 48, 49]. The introduction of the STP has provided novel insights into the topic, converting the identification problem of the original system into identifying its algebraic forms, specifically into state and output structure matrices. Initial studies, [50], employed this method to establish conditions for identifying BNs from observed data and provided identification algorithms for some particular network types. Concurrently, [51] demonstrated that BCNs could be identified by satisfying observability and controllability criteria. Subsequently, [52] addressed the identification of B(C)Ns without stringent system property requirements. Zhang [53] optimized the identification process of the relationship between structure matrices and nodes in [50] by incorporating prior topological information, while [54] further leveraged controllability and observability theories to design efficient identification algorithms. Additional challenges in identifying delayed B(C)Ns and singular B(C)Ns were also explored [55, 56].

Observability is a key prerequisite for identification [51, 54], which can be viewed as a measure of the distinguishability between distinct initial states of a system using only input-output data [57]. That is, it concerns whether the initial state can be uniquely determined from input-output data. Cheng [34] first introduced the concept of observability of BNs and established corresponding criteria. [58] proposed a concept different from that in [34], provided the necessary and sufficient conditions for the observability of the system, and established a method for observer design. For a comprehensive introduction of various observability definitions for BNs, please refer to [59].

Subsequent studies have further developed various criteria and methods for observability analysis of B(C)Ns. Zhao introduced an input-state incidence matrix for observability determination [60]. [61] proposed a graph-theoretic approach to transform observability into a structural property of graphs. Zhang further introduced finite automata theory and proposed the concept of weighted pairing graphs, successfully unifying the four mutually inequivalent BCN observability definitions in the literature and revealing the nonlinear nature of the systems [62]. Yu then gave the necessary and sufficient conditions for observability from the perspective of logical matrix equations [63] and further provided a more general and direct criterion for observability of BNs [64], while [65] skillfully transformed the observability problem into a set controllability problem, achieving theoretical transformation and unifi-

cation. Additionally, the observability of other logical networks, including SBCNs [66, 67], delayed BCNs [68, 69], and probabilistic BNs [70], has also been investigated. Existing observability studies on SBCNs generally assume that the switching signal affects only the state transition dynamics [66, 67]. In contrast, for SBNs in which switching signals affect both the state transition and output dynamics, switching signals may introduce additional observational information, enabling the distinguishability of all states more easily. To the best of our knowledge, the such systems' observability has received very limited attention. Zhang et al. [12] investigated the observability of this class of SBNs with control inputs by verifying the recognizable languages of deterministic finite automata.

Reachability and controllability are two fundamental properties of logical dynamical systems, which also play a crucial role in system analysis and identification. Reachability characterizes whether a given state can be driven from another state via admissible inputs. Controllability is a stronger global attribute, which refers to the ability of the system to reach any target state through appropriate control input sequences from any initial state in the state space [34]. Zhao proposed controllability criteria and a controller design algorithm through the input-state incidence matrix [60]. [71] introduced the Perron-Frobenius theory to provide a profound mathematical perspective for controllability determination. Subsequently, [7] further solved the controllability problem of SBCNs. Further, [72] investigated reachability and controllability for more complex delayed SBCNs. Moreover, [73] unified the judgment criteria under mixed controls through the results of set controllability. Aiming at the randomness existing in real systems, [74] extended the theoretical framework of the Markov jump model. Recently, the focus of research has shifted to addressing the computational bottleneck of large-scale networks, and aggregation methods and reinforcement learning methods have been introduced to solve the problem [75, 76].

In actual biological systems, many key processes (such as apoptosis or differentiation) often behave as the state trajectory entering a specific functional region rather than a single state, from which the two important concepts of set reachability and set controllability are derived. Set reachability refers to the ability of the system trajectory to enter the target set from a given initial state (or set) in a finite number of steps; in the controlled case, if there is a control sequence that can drive the system from any initial state to the target set, the system is said to be set controllable. Cheng first revealed the intrinsic equivalence between observability and set controllability of BCNs, laying an algebraic foundation for the determination of this property [65]. Subsequently, researchers have made extensive extensions

for more complex network models. For example, [77] studied set controllability for SBCNs; [78] studied Markov jump BNs, defined strong/weak set reachability, and gave probabilistic criteria; [79] and [80] respectively explored the relevant properties of impulsive probabilistic BCNs and state-dependent impulsive systems, the latter also optimized the design of pinning controllers through Hamming distance. In addition, for special update rules and structures, [81] analyzed set reachability under state-dependent asynchronous updates; [82] solved the set controllability problem of periodically switched SBCNs.

The study of set reachability and set controllability has formed a complete system from basic theory to complex dynamic models. These properties not only form the cornerstone of the qualitative analysis of logical dynamical systems, but also provide a core theoretical foundation for the further study of advanced control problems such as observability and reconstructibility, set stabilization, and output tracking [65, 83, 84, 85]. By accurately depicting the set evolution rule, we can further realize the dynamic control and global optimization of individual strategic behavior under the complex framework of NEGs. This series of theoretical results provides key analysis tools and criteria for subsequent research on output tracking control and game strategy optimization under complex logical networks.

1.2.2 Output Tracking Control of BNs and SBNs

The output tracking problem (OTP) aims to seek proper controllers to ensure that the outputs of a controlled system always track a prescribed reference signal after some time. It is a fundamental issue in control theory. The OTP of BCNs with respect to a fixed signal was first discussed in [86]; the authors deduced solvability criteria using the optimal control method, and then [87] investigated it using the state feedback method. Subsequent studies approached the OTP of BCNs with a dynamic signal using bilinear equations [88], and control invariant subsets of auxiliary systems [85]. Additional studies have explored the OTP of other types of BNs, such as SBCNs [89, 90], delayed BCNs [91], and probabilistic BNs [92].

Notably, to overcome the computational complexity inherent in traditional methods, [93] investigated the output trackability of BCNs by employing the Ledley antecedence solution, which provides a more direct way to solve logical equations. This approach naturally leads to the introduction of the truth matrix method [94]. The truth matrix, originating from the algebraic representation of logical mappings, serves as a powerful tool to describe the input-output relationships of logical operators. Furthermore, the truth matrix approach provides a clear description of

one-step evolutionary dynamics while maintaining computational simplicity, and it can be readily implemented via software tools such as MATLAB. By leveraging the properties of truth matrices, a variety of control issues beyond OTP have been addressed, such as stabilization [95, 96], stabilization [97], optimal control of BNs [98], and strategy optimization of NEGs [99], offering a more efficient computational framework for logical systems.

While achieving output tracking (OT) is highly desirable, it is often challenging in practice due to resource constraints and the need for improved system performance. In many real-time control applications, systems must respond to periodic signals or events. For example, in automation equipment operations, particularly on assembly lines or in manufacturing processes, machines repeat a series of precise actions, often periodically, such as picking and placing components during the assembly cycle. Their control systems must track these periodic reference signals to ensure accuracy and repeatability in operations [100]. Similarly, in power systems, the output of a generator must be adjusted to specific frequencies and voltages to ensure the stable operation of the power grid and a reliable energy supply [101]. Periodic OT, which aims to design controllers such that the system tracks a periodic reference signal, has consequently gained widespread attention [102, 103, 104].

Despite its attractive properties, periodic OT may be infeasible or even lead to malfunctioning in certain real-time control applications. For instance, in a fault-tolerant environmental monitoring network, sensors may intermittently fail due to hardware faults or environmental disruptions, causing temporary deviations from the desired reporting pattern. However, backup mechanisms or system redundancies can enable recovery [105], ensuring that the outputs intermittently and infinitely align with the reference signal. In such scenarios, conventional OT would classify these deviations as system failures, as continuous alignment with the reference signal is not maintained. Similarly, in gene therapy for cancer or Alzheimer's disease, therapeutic gene expression must be carefully regulated to achieve desired expression levels intermittently, though not necessarily periodically. This approach mitigates the risks of overexpression while maintaining long-term therapeutic efficacy [106]. To the best of our knowledge, there is currently no concept to describe such tracking phenomenon.

1.2.3 Optimization Control of NEGs

NEGs provide a powerful framework for analyzing the collective behavior of interacting individuals in biological, social, and economic systems. However, due to

the lack of corresponding mathematical tools, it is difficult to directly analyze the dynamics of NEGs. Luckily, by allowing the multiplication of matrices with arbitrary dimensions, STP enables the representation of strategy profiles as canonical basis vectors, through which NEGs can be formulated as logical dynamical systems. Cheng [107] systematically analyzed the existence of stable strategy situations of NEGs for the first time, and studied the equivalence of two NEGs by using equivalence classes. This algebraic representation has led to significant progress in the modeling, stability analysis, consensus, and control of NEGs [108, 109, 110]. A comprehensive overview of the applications of the STP method in NEGs has been reported in [111].

In fact, many real-world interactions are inherently higher-order [112, 113]; thus, researchers have shifted their attention from NEGs to HNEGs. Existing studies on HNEGs, such as [114, 115, 116], were mainly developed under the assumption of homogeneous hypergraphs. In particular, [116] employed the STP method to derive a fundamental evolutionary equation for all players, thereby establishing a mathematical model for HNEGs. This framework has facilitated subsequent investigations into dynamical behaviors, including convergence and potential properties [117, 118, 119]. However, real-world hyper-networks are typically heterogeneous, exhibiting both degree and order heterogeneity, which significantly complicates the modeling and analysis. For example, [24, 25] discussed how the heterogeneity of degree and order affects the evolution of group cooperation behavior. Although some works, such as [120], have considered heterogeneous hypergraphs, the payoff of each player is still constructed as the aggregation of pairwise interactions within hyperedges.

Strategy optimization is a fundamental objective in evolutionary game analysis, aiming to enhance group payoffs, foster the emergence of cooperation, or regulate collective behaviors. Existing literature has primarily investigated convergence toward optimal strategies, such as Nash equilibria, under unconstrained scenarios [121, 122, 123]. However, such approaches emphasize optimality while overlooking practical survival requirements of individual players. In many real-world scenarios, players must maintain their payoffs above certain thresholds to sustain participation, such as firms maintaining operational viability or individuals remaining active in social or economic systems [124, 125, 126]. From this perspective, ensuring that each player's payoff reaches and remains above a prescribed threshold is crucial for system stability and sustainability.

1.3 From Critical Gaps to Thesis Contributions

The existing literature underscores the profound significance of identifying and controlling logical dynamic systems, as well as optimizing strategies for NEG_s. However, a critical theoretical disconnect persists within current control and optimization paradigms, which motivates the core research of this thesis.

First, when switching signals modulate both the state evolution and the output mapping, the input-output behavior of SBNs becomes more complex, as identical state trajectories under different subnetworks may produce distinct outputs. However, existing studies have ignored the impact of switching signals on system identification and the identification methods used for BCN identification (e.g., [54]) are generally not applicable under such conditions. Moreover, an SBN can be viewed as a collection of BNs evolving under a switching signal. This raises the natural question: for an SBN, is there any relationship between the identification of the subnetworks and that of the entire system? Therefore, focusing on such SBNs, we establish necessary and sufficient conditions for system identifiability by establishing the exact correspondence between system states and outputs by incorporating temporal information and develop the corresponding identification algorithms. And we find that even if certain subnetworks of an SBN are unidentifiable, the overall SBN may still be identifiable.

Additionally, for the observability of SBNs where the switching signal influences both states and outputs, existing study relies on the recognizable languages of the deterministic finite automaton. However, the applicability of this method is limited to very small-scale systems and requires knowledge of formal languages and deterministic finite automata to understand the approach [127]. Thus we propose an alternative algebraic criterion based solely on the structural matrices of SBNs, which can be straightforwardly implemented via matrix operations.

Then, although OT and periodic OT are important research topics in the field of control, they cannot capture the infinite and intermittent tracking phenomenon of a system. This observation motivates the notion of recurrent output tracking (ROT), which relaxes the requirements of OT and periodic OT. In this thesis, we first introduce the concept of ROT for BNs and BCNs starting from a given initial state, propose the algebraic conditions for solvability of the recurrent output tracking problem (ROTP), and give constructive methods to design the corresponding state feedback controllers. As an extension, we further investigate the ROT for SBCNs starting from any initial state, and the obtained results highlight the interplay between switching signals and controls in achieving ROT.

Finally, existing studies mainly focus on homogeneous HNEGs or HNEGs whose hyperedges are primarily utilized to describe participation structures rather than to induce genuine multi-player payoff coupling. It is, therefore, imperative to investigate heterogeneous HNEGs characterized by hyperedge-induced multi-player payoff structures. Moreover, while threshold constraints have been widely discussed in traditional pairwise interaction settings, they remain largely unexplored in the context of heterogeneous HNEGs, where structural heterogeneity introduces unprecedented analytical and control challenges. To address these critical gaps, we establish a rigorous algebraic framework for such heterogeneous HNEGs via the STP method. Building upon this framework, we introduce a controlled heterogeneous HNEG model by incorporating external control players into the game dynamics. We derive necessary and sufficient conditions to guarantee that all players' payoffs simultaneously reach and remain above a prescribed threshold. Furthermore, a time-optimal state feedback controller is designed to achieve this control objective.

1.4 Main Contents and Thesis Organization

In reaction to the above analysis, this thesis investigates the identification of SBNs, the ROT of BNs and SBCNs, and the strategy optimization of heterogeneous HNEGs. The main contents of each chapter are summarized as follows:

- **Chapter 1: Introduction** This chapter introduces the background and motivation of this thesis, including BNs, SBNs, and HNEGs, and reviews the related literature. It formulates the main research problems and provides a systematic overview of existing studies, highlighting key concepts and methodologies. In addition, the chapter critically analyzes related work, identifies research gaps, and establishes the positioning of this research within the broader academic landscape.
 - **Chapter 2: Preliminaries** This chapter introduces the necessary preliminaries, including the STP tool, BNs, SBNs, and basic concepts in NEGs on hypergraphs.
 - **Chapter 3: Identification of Switched Boolean Networks** This chapter focuses on the identification problem for SBNs under general switching structures where the switching signal may influence both the state transition and the output mapping. By establishing a unified algebraic representation,
-

necessary and sufficient conditions for system identifiability are derived. Furthermore, both single-sample and multiple-sample identification schemes are developed, revealing that the overall system can remain identifiable even when certain subnetworks are not individually identifiable.

- **Chapter 4: Recurrent Output Tracking of Boolean Networks** A novel control objective, termed ROT, is introduced for BNs and BCNs. This formulation relaxes the requirements of conventional OT by allowing the system outputs to track the desired reference signal infinitely often over time. Algebraic conditions for solvability are established, and constructive methods are proposed to design state feedback controllers that guarantee recurrent tracking performance.
- **Chapter 5: Recurrent Output Tracking of Switched Boolean Control Networks** The ROTP is further extended to SBCNs, where both control inputs and switching signals are treated as design variables. By reformulating the problem in terms of set controllability, equivalent conditions for solvability are derived, and systematic controller design procedures are developed.
- **Chapter 6: Strategy Optimization of Controlled Heterogeneous Hypernetworked Evolutionary Games** The proposed algebraic framework is applied to heterogeneous HNEGs. By representing the game dynamics as logical dynamical systems, the strategy optimization problem is formulated as a minimum-time set controllability problem. Conditions for the existence of admissible control strategies are established, and time-optimal strategy design methods are developed under payoff threshold constraints.
- Finally, **Chapter 7: Conclusion and Future Work** This chapter concludes the thesis and discusses possible directions for future research.

The structure of the thesis and the summary of contributions are shown in Figures 1.1 and 1.2, respectively.

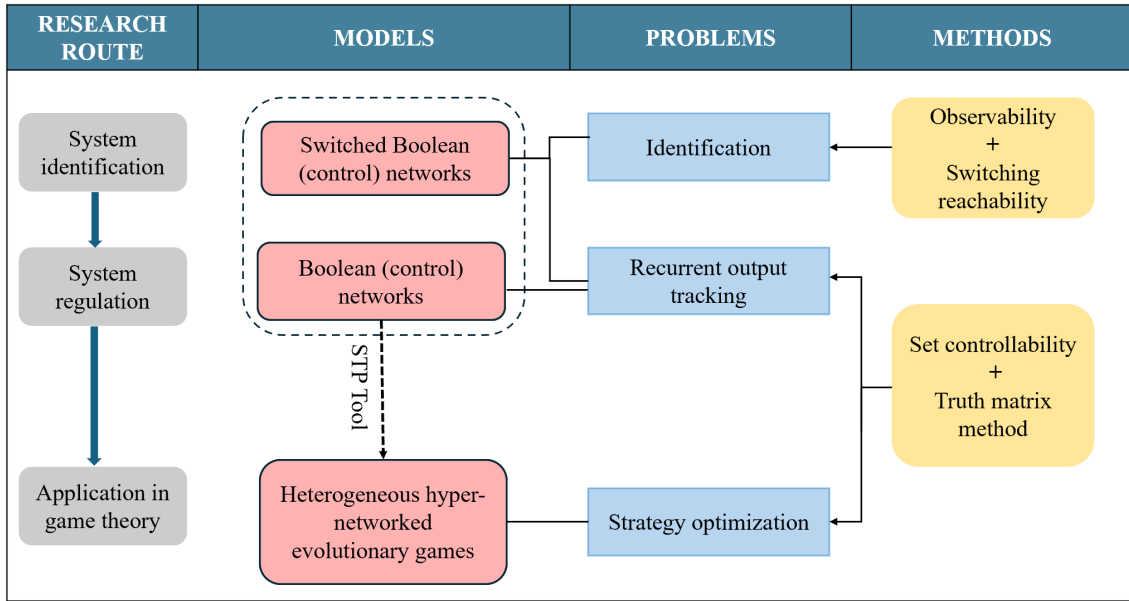


Figure 1.1: The structure of the thesis.

Chapter	Main Assumptions	Key Theorems	Proposed Algorithms	Validation Examples	Publications
Chapter 3 Identification and Observability of SBNs	- Switching output data is sufficient - Data corresponding to all possible initial state can be acquired under multi-sample	- T3.9: Algebraic criterion for observability - T3.14 and T3.17: Necessary and sufficient criteria for system identifiability	A1: Checking observability of SBNs A2 and A3: Structural identification algorithms under the single sample and multi-sample scenarios	E3.19 and E3.20: Identification under the single sample and multi-sample scenarios	Information Sciences (Published)
Chapter 4 Recurrent Output Tracking of BNs	Reference signal is constant	- T4.6: Algebraic solvability criteria for ROT of BNs - T4.7: Algebraic solvability criteria for ROT of BCNs	A5 and A6: The sets consisting in all states whose outputs can recurrently track the desired reference signal for BNs and BCNs A7: state feedback controller design for the ROTP	E4.13: A BN proposed in reference [32] E4.14: A modified model based on the lac operon in the bacterium <i>Escherichia coli</i>	Nonlinear Analysis: Hybrid Systems (Published)
Chapter 5 Recurrent Output Tracking of SBCNs	Reference signal is constant	T5.8: Necessary and sufficient conditions for global ROT of SBCNs	A8: Switching signal and control design for the ROTP of SBCNs	E5.11: A modified model based on the lac operon in the bacterium <i>Escherichia coli</i>	2025 American Control Conference (Published)
Chapter 6 Strategy Optimization of Heterogeneous HNEGs	- All players share a common finite strategy set - The strategy update rule: myopic best response adjustment rule - The first several players act as control players	- T6.8: Necessary and sufficient conditions for a control invariant subset - T6.11 and T6.12: Existence criteria for minimum-time state feedback controllers	A9: Computation of the maximum control invariant subset A10: state feedback controller design for the strategy optimization	E6.15: A stag hunt game on a heterogeneous hypergraph consisting of six nodes and six hyperedges	IEEE Transactions on Control of Network Systems (Under Review)

Figure 1.2: The summary of thesis contributions.

Chapter 2

Preliminaries

In this chapter, we will recall the definition and properties of STP, algebraic expression of logical functions, B(C)Ns, SB(CN)s, and NEGs and hypergraphs, which will be used throughout this thesis.

2.1 Semi-tensor Product of Matrices

2.1.1 Definition and Basic Properties

Before presenting the definition of the STP, we first introduce the concept of the Kronecker product of matrices.

Definition 2.1. ([30]) Let $A = (a_{ij}) \in \mathbb{R}_{m \times n}$ and $B = (b_{ij}) \in \mathbb{R}_{p \times q}$. The *Kronecker product* of matrices A and B is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix} \in \mathbb{R}_{mp \times nq}. \quad (2.1)$$

Definition 2.2. ([30]) The semi-tensor product (STP) of two matrices $A \in \mathbb{R}_{m \times n}$ and $B \in \mathbb{R}_{p \times t}$ is defined as

$$A \ltimes B = (A \otimes I_{\frac{lcm(n,p)}{n}})(B \otimes I_{\frac{lcm(n,p)}{p}}).$$

If $n = p$ in Definition 2.2, then $A \ltimes B = AB$. Thus the STP generalizes the conventional matrix product while preserving its main properties such as associative

law and distributive law. For simplicity, the symbol “ $\ltimes$ ” is omitted throughout the thesis whenever no ambiguity arises.

Lemma 2.3. ([30]) *Some basic properties of STP are listed as follows:*

(i) For $X \in \mathbb{R}^t$ and $M \in \mathbb{R}_{m \times n}$,

$$XM = (I_t \otimes M)X.$$

(ii) For $X \in \mathbb{R}^t$, $Y \in \mathbb{R}^k$,

$$W_{[t,k]}XY = YX,$$

where $W_{[t,k]} = [I_k \otimes \delta_t^1, I_k \otimes \delta_t^2, \dots, I_k \otimes \delta_t^t] \in \mathcal{L}_{kt \times kt}$ is called a swap matrix.

(iii) For $X \in \Delta_t$,

$$XX = M_{r,t}X,$$

where $M_{r,t} = \text{diag}\{\delta_t^1, \delta_t^2, \dots, \delta_t^t\}$ is called a power-reducing matrix.

(iv) For $X \in \Delta_{k_1}$ and $Y \in \Delta_{k_2}$, define the following two dummy matrices

$$D_f^{[k_1, k_2]} = I_{k_1} \otimes \mathbf{1}_{k_2}^T, \quad D_r^{[k_1, k_2]} = \mathbf{1}_{k_1}^T \otimes I_{k_2}.$$

Then $D_f^{[k_1, k_2]}XY = X$ and $D_r^{[k_1, k_2]}XY = Y$, where $D_f^{[k_1, k_2]}$ and $D_r^{[k_1, k_2]}$ are called “front-maintaining operator” and “rear-maintaining operator” respectively.

For more details on the STP, the reader is referred to [30].

2.1.2 Algebraic Representation of Logical Functions

Throughout this thesis, the semi-tensor product provides a unified algebraic representation for all logical systems under consideration.

Identify “ $1 \sim \delta_2^1$ ” and “ $0 \sim \delta_2^2$ ”, δ_2^1 and δ_2^2 are called the vector form of 1 and 0 respectively. Then it is easy to transform a Boolean function into an algebraic form by the following lemma.

Lemma 2.4. ([30]) *Let $f(X_1, X_2, \dots, X_n) : \mathcal{D}^n \rightarrow \mathcal{D}$ be a Boolean function. Then a unique matrix $M_f \in \mathcal{L}_{2 \times 2^n}$ can be found, called the structure matrix of f , such that*

$$f(X_1, X_2, \dots, X_n) = M_f \ltimes_{i=1}^n x_i,$$

where $x_i \in \Delta$ is the vector form of X_i , $i = \{1, 2, \dots, n\}$.

For any $i \in \mathcal{D}_k$, it is associated with the corresponding canonical vector δ_k^i , namely, $i \sim \delta_k^i$, $i = 1, 2, \dots, k$. Accordingly, i can be equivalently viewed as an element of Δ_k , which refers to the vector form of i . Then it is easy to transform a k -valued logical function into an algebraic form by the following lemma.

Lemma 2.5. [30] *Let $f : \mathcal{D}_k^n \rightarrow \mathcal{D}_k$ be a k -valued logical function, then there exists a unique matrix $N_f \in \mathcal{L}_{k \times k^n}$, such that in vector form we have*

$$f(X_1, \dots, X_n) = N_f \bowtie_{i=1}^n x_i, \quad (2.2)$$

where M_f is called the structure matrix of f . Additionally, let $c : \mathcal{D}_k^n \rightarrow \mathbb{R}$ be a k -valued pseudo-logical function, then there exists a unique row vector $V^c \in \mathbb{R}^{k^n}$, such that in vector form we have

$$c(X_1, \dots, X_n) = V^c \bowtie_{i=1}^n x_i. \quad (2.3)$$

where V^c is called the structure vector of c .

The algebraic representations of the STP and logical functions introduced in this section form the mathematical foundation of the entire thesis. By transforming discrete logical variables and mappings into vector and matrix operations, STP eliminates the essential barriers of analytic logical dynamics systems. In the following chapters, this algebraic framework will be directly used to construct the evolution equations of various complex networks, especially the structure matrix derivation of BNs in Chapter 3 and the explicit establishment of HNEG dynamics equations in Chapter 6.

2.2 Algebraic Representation of BNs and BCNs

A BN is described as:

$$\begin{cases} X_i(t+1) = f_i(X_1(t), X_2(t), \dots, X_n(t)), & i = 1, 2, \dots, n, \\ Y_j(t) = h_j(X_1(t), X_2(t), \dots, X_n(t)), & j = 1, 2, \dots, p, \end{cases} \quad (2.4)$$

where $X_i \in \mathcal{D}$, $i = 1, 2, \dots, n$ denotes the state node, $Y_j \in \mathcal{D}$, $j = 1, 2, \dots, p$ denotes the output node, and $f_i : \mathcal{D}^n \rightarrow \mathcal{D}$, $h_j : \mathcal{D}^n \rightarrow \mathcal{D}$ are Boolean functions. In what follows we use $X(t) := (X_1(t), X_2(t), \dots, X_n(t))$ and $Y(t) := (Y_1(t), Y_2(t), \dots, Y_p(t))$.

Let $X_i \sim x_i$ and $Y_j \sim y_j$. According to Lemma 2.4, system (2.4) can be equiva-

lently expressed as:

$$\begin{cases} x_i(t+1) = L_i \ltimes_{i=1}^n x_i(t), & i = 1, 2, \dots, n, \\ y_j(t) = H_j \ltimes_{i=1}^n x_i(t), & j = 1, 2, \dots, p, \end{cases} \quad (2.5)$$

where $L_i \in \mathcal{L}_{2 \times 2^n}$ is the structure matrix of function f_i and $H_j \in \mathcal{L}_{2 \times 2^n}$ is the structure matrix of function h_j . Denote $x(t) = \ltimes_{i=1}^n x_i(t) \in \Delta_{2^n}$ and $y(t) = \ltimes_{j=1}^p y_j(t) \in \Delta_{2^p}$, then system (2.5) can be written as:

$$\begin{cases} x(t+1) = Lx(t), \\ y(t) = Hx(t), \end{cases} \quad (2.6)$$

where $L := L_1 * L_2 * \dots * L_n \in \mathcal{L}_{2^n \times 2^n}$ and $H := H_1 * H_2 * \dots * H_p \in \mathcal{L}_{2^p \times 2^n}$ are the state structure matrix and the output structure matrix of system (2.4), respectively.

By adding control to the BN (2.4), it becomes a BCN, which is described as:

$$\begin{cases} X_i(t+1) = f_i(X(t), U_1(t), U_2(t), \dots, U_m(t)), & i = 1, 2, \dots, n, \\ Y_j(t) = h_j(X(t)), & j = 1, 2, \dots, p, \end{cases} \quad (2.7)$$

where $X_i \in \mathcal{D}$, $i = 1, 2, \dots, n$ denotes the state node, $U_k \in \mathcal{D}$, $k = 1, 2, \dots, m$ denotes the input node, $Y_j \in \mathcal{D}$, $j = 1, 2, \dots, p$ denotes the output node, and $f_i : \mathcal{D}^{n+m} \rightarrow \mathcal{D}$, $h_j : \mathcal{D}^n \rightarrow \mathcal{D}$ are Boolean functions. In what follows we use $U(t) := (U_1(t), U_2(t), \dots, U_m(t))$.

Let $U_k \sim u_k$. Denote $u(t) = \ltimes_{k=1}^m u_k(t) \in \Delta_{2^m}$. Similar to the algebraic expression of system (2.4), system (2.7) can be equivalently expressed as according to Lemma 2.4:

$$\begin{cases} x(t+1) = Lu(t)x(t), \\ y(t) = Hx(t), \end{cases} \quad (2.8)$$

where $L \in \mathcal{L}_{2^n \times 2^{n+m}}$ and $H \in \mathcal{L}_{2^p \times 2^n}$.

This section demonstrates how STP can be used to transform traditional BNs and BCNs into linear discrete-time dynamical systems. This algebraic form (as shown in equations (2.6) and (2.8)) greatly facilitates the research in Chapter 4

2.3 Algebraic Representation of SBNs and SBCNs

A switched Boolean network (SBN) with μ subnetworks is modeled as:

$$\begin{cases} X_i(t+1) = f_i^{\Sigma(t)}(X(t)), & i = 1, 2, \dots, n, \\ Y_j(t) = h_j(X(t)), & j = 1, 2, \dots, p, \end{cases} \quad (2.9)$$

where $\Sigma(t) : \mathbb{N} \rightarrow \{1, 2, \dots, \mu\}$ is the switching signal, $X_i \in \mathcal{D}$, $i = 1, 2, \dots, n$ denotes the state node, $Y_j \in \mathcal{D}$, $j = 1, 2, \dots, p$ denotes the output node, and $f_i^{\Sigma(t)} : \mathcal{D}^n \rightarrow \mathcal{D}$, $h_j : \mathcal{D}^n \rightarrow \mathcal{D}$ are Boolean functions.

Relying on Lemma 2.4, the equivalent algebraic form of system (2.9) is:

$$\begin{cases} x(t+1) = L^{\Sigma(t)}x(t), \\ y(t) = Hx(t), \end{cases} \quad (2.10)$$

where $x(t) = \times_{i=1}^n x_i(t) \in \Delta_{2^n}$, $y(t) = \times_{j=1}^p y_j(t) \in \Delta_{2^p}$, $L^k \in \mathcal{L}_{2^n \times 2^n}$ is the state structure matrices of the k -th subnetwork and $H \in \mathcal{L}_{2^p \times 2^n}$ the state structure matrices of the system.

Set $L = [L^1, L^2, \dots, L^\mu] \in \mathcal{L}_{2^n \times \mu \cdot 2^n}$. Let $\Sigma \sim \sigma$, where $\sigma(t) \in \Delta_\mu$. Then, SBN (2.9) is equivalent to

$$\begin{cases} x(t+1) = L\sigma(t)x(t), \\ y(t) = Hx(t), \end{cases} \quad (2.11)$$

Obviously, $L\sigma(t) = L^{\Sigma(t)}$.

Remark 2.6. If we regard the switching signals as controls for the SBN (2.9) the system (2.9) will be equivalent to the BCN (2.7).

By adding control to the above SBN, system (2.9) becomes an SBCN with μ subnetworks, which is described as:

$$\begin{cases} X_i(t+1) = f_i^{\Sigma(t)}(X(t), U(t)), & i = 1, 2, \dots, n, \\ Y_j(t) = h_j(X(t)), & j = 1, 2, \dots, p, \end{cases} \quad (2.12)$$

where the function $f_i^{\Sigma(t)} : \mathcal{D}^{m+n} \rightarrow \mathcal{D}$ and $h_j : \mathcal{D}^n \rightarrow \mathcal{D}$ are Boolean functions.

Similar to the algebraic expression of the system (2.9), the SBCN (2.12) can be equivalently expressed as:

$$\begin{cases} x(t+1) = L\sigma(t)u(t)x(t), \\ y(t) = H\sigma(t)x(t), \end{cases} \quad (2.13)$$

where $L \in \mathcal{L}_{2^n \times \mu \cdot 2^{n+m}}$ and $H \in \mathcal{L}_{2^p \times \mu \cdot 2^n}$.

The algebraic representations of traditional SBNs and SBCNs further extend the modeling scope of BNs, enabling effective description of dynamic systems with different subnetworks. The switched models derived in this section lays the foundation for the SBNs in Chapter 3, where switching signals simultaneously affect both state and output, and becomes the core research object of Chapter 5.

2.4 NEGs and Hypergraphs

Definition 2.7. ([107]) A finite normal game $G = (N, \mathcal{S}, C)$ consists of the following elements:

- (i) The set of players: $N = \{1, 2, \dots, n\}$.
- (ii) The strategy profile: $\mathcal{S} = \prod_{i=1}^n \mathcal{S}_i$, where $\mathcal{S}_i = \{1, 2, \dots, k_i\}$ is the set of strategies of player i . Additionally, $\mathcal{S}_{-i} := \prod_{j \neq i} \mathcal{S}_j$.
- (iii) The set of payoff functions: $C = \{c_1, c_2, \dots, c_n\}$, where $c_i : \mathcal{S} \rightarrow \mathbb{R}$ denotes the payoff function of player i .

The finite normal game representation serves as a fundamental building block for modeling multi-player interactions on hyperedges in heterogeneous HNEGs.

Definition 2.8. A normal game with m players is called an m -player fundamental network game (FNG), if $\mathcal{S}_1 = \mathcal{S}_2 = \dots = \mathcal{S}_m = \{1, 2, \dots, k\}$.

We then introduce the definition of a hypergraph and related heterogeneity notions.

Definition 2.9. ([21, 24])

- 1) The tuple (N, E) is called a hypergraph if it consists of a set of nodes $N = \{1, 2, \dots, n\}$ and a set of hyperedges $E = \{E^1, E^2, \dots, E^r\} \subseteq 2^N$, where 2^N is the set of subsets of set N . The group size of the l -th hyperedge is referred to as its order, denoted by $g_l := |E^l|$. The number of hyperedges to which node i belongs is called the degree of node i , denoted by k_i .
- 2) If there exist at least two nodes i and j such that $k_i \neq k_j$, then (N, E) is called a degree-heterogeneous hypergraph. If there exist $l_1, l_2 \in \{1, 2, \dots, r\}$ such that $g_{l_1} \neq g_{l_2}$, then (N, E) is called an order-heterogeneous hypergraph. A hypergraph that is both degree-heterogeneous and order-heterogeneous is called a heterogeneous hypergraph.

In particular, when $g_l = 2$ for all $l = 1, \dots, r$, the hypergraph (N, E) reduces to a conventional graph. A simple example of a heterogeneous hypergraph and a graph is given below.

Example 2.10. A heterogeneous hypergraph with five nodes $N = \{1, 2, 3, 4, 5\}$, three hyperedges $E = \{E^1, E^2, E^3\}$ is shown in Figure 2.1a). Here, $E^1 = \{2, 3\}$, $E^2 = \{3, 4, 5\}$, and $E^3 = \{1, 3, 4\}$. The corresponding hyperedge orders are $g_1 = 2$ and $g_2 = g_3 = 3$. The degrees of the five nodes are $k_1 = k_2 = k_5 = 1$, $k_3 = 3$ and $k_4 = 2$. As illustrated in Figure 2.1b), a traditional graph representation reduces each hyperedge to pairwise edges, thereby losing the explicit description of higher-order interactions.

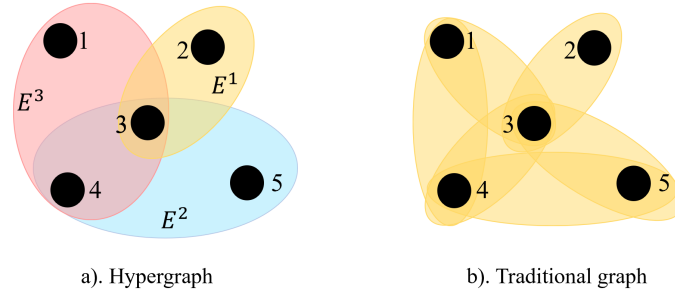


Figure 2.1: Hypergraph and graph. a). A heterogeneous hypergraph with five nodes and three hyperedges; b). A graph with five nodes and six edges.

In games on the conventional graph, the overall payoff of player i is typically obtained by summing the payoffs from all 2-player FNGs over its neighborhood [107, 108, 109, 110]. However, such a pairwise decomposition is no longer sufficient to capture genuine multi-player interactions. Based on the heterogeneous hypergraph structure, each hyperedge $E^l \in E$ naturally induces a normal game among the participating players. Specifically, for a hyperedge $E^l = \{i_1, i_2, \dots, i_m\}$, the corresponding interaction can be formulated as an m -player FNG, where the payoff function of each player is denoted by $c_i^{E^l}$, $i \in E^l$. It is noted that hyperedges of the same order share the payoff structures in this paper. Suppose that the heterogeneous hypergraph admits α hyperedge orders, denoted by $\bar{g} := \{\bar{g}_1, \bar{g}_2, \dots, \bar{g}_\alpha\}$. Thus, there are α FNGs, denoted by $G_1, G_2, \dots, G_\alpha$, where G_m is a $\bar{g}_m$ -player FNG, on the hypergraph.

For a player $i \in N$, let k_i hyperedges on which it belongs be denoted by $E(i) := \{E_i^1, E_i^2, \dots, E_i^{k_i}\}$. The set of neighbors of player i on the hyperedge E_i^l is denoted

by $N(i, E_i^l)$. Then, the overall neighbor set of player i is given by

$$N_i := \bigcup_{l=1}^{k_i} N(i, E_i^l).$$

Definition 2.11. A heterogeneous hyper-networked evolutionary game (HNEG) is described by a triple $((N, E), G, \Pi)$, where

- (i) A heterogeneous hypergraph (N, E) , where $N = \{1, 2, \dots, n\}$ is the set of players (nodes) and $E = \{E^1, E^2, \dots, E^r\}$ is the set of hyperedges.
- (ii) The set of FNGs $G := \{G_1, G_2, \dots, G_\alpha\}$, where G_m is a $\bar{g}_m$ -player FNG, $m \in \{1, 2, \dots, \alpha\}$. It means that if $E^l = \{i_1, i_2, \dots, i_{\bar{g}_m}\}$, then players $i_1, i_2, \dots, i_{\bar{g}_m}$ play the $\bar{g}_m$ -player FNG.
- (iii) The strategy updating rule Π , which decides the strategy of player i at time $t + 1$ and is expressed as

$$x_i(t + 1) = f_i(\{x_j(t), c_j(t) \mid j \in N_i\}), \quad i \in N, \quad (2.14)$$

where $x_i(t) \in \mathcal{S}_i$ denotes the strategy of player i at time t , N_i denotes the set of neighbors for player i , $c_j(t)$ denotes the payoff of player $j \in N_i$ at time t and the strategy updating function f_i is a mapping from the strategies and payoffs of the players in N_i at time t to a strategy.

Remark 2.12. Definition 2.11 provides a more general concept of the heterogeneous HNEGs. The public goods games on hypergraphs studied in [17, 24, 25] and the multiplayer snowdrift games on hypergraphs studied in [25] are particular instantiations of the proposed heterogeneous HNEG framework.

This section introduces the basic definitions of heterogeneous hypergraphs and HNEGs. Compared to traditional graphs, hypergraphs can more realistically depict high-order nonlinear interactions between multiple players, but also significantly increase the mathematical complexity of revenue calculation and strategy updates. The game structure and neighbor topology defined in this section will be combined with the STP tool in the first section and used as the core tool in Chapter 6. At that time, we will overcome the modeling difficulties caused by high-order interactions, formally establish an algebraic framework for heterogeneous HNEG, which can be viewed as a class of logical dynamical systems, and solve the optimization problem based on this framework.

Chapter 3

Identification of Switched Boolean Networks

System identification plays a fundamental role in inferring system behaviors from observed data and serves as a prerequisite for subsequent analysis and control design. To date, identification problems have been extensively studied for BNs and BCNs [50, 51, 52, 53, 54]. For switched systems, it is commonly assumed that the switching signal affects only the state transition dynamics, while the output mapping remains invariant. Under this assumption, the identification of SBNs can be reduced to that of BCNs. However, in many practical scenarios, the switching mechanism may simultaneously influence both the state evolution and the output generation processes [11, 12]. In this more general setting, identical state trajectories under different subnetworks may produce distinct outputs, which significantly increases the complexity of the identification problem and renders existing methods inapplicable.

To address these challenges, this chapter investigates the identification problem for SBNs under general switching structures. By employing the STP method, a unified algebraic representation of the SBN is established. Based on this formulation, the reachability and observability properties of SBNs are first analyzed, and corresponding detection conditions are derived. Building on these results, identification criteria are developed for both single-sample and multiple-sample scenarios. By incorporating temporal information, the correspondence between states and outputs can be determined, leading to effective identification algorithms. It is further shown that even if some subnetworks are unidentifiable, the overall SBN may still be identifiable.

3.1 Problem Formulation

Consider the SBN (2.9) in Chapter 2. When the switching signals also affect the outputs of the μ subsystems, the mathematical model of SBN (2.9) becomes the following form:

$$\begin{cases} X_i(t+1) = f_i^{\Sigma(t)}(X(t)), & i = 1, 2, \dots, n, \\ Y_j(t) = h_j^{\Sigma(t)}(X(t)), & j = 1, 2, \dots, p, \end{cases} \quad (3.1)$$

where $X_i \in \mathcal{D}$, $i = 1, 2, \dots, n$ denotes the state node, $Y_j \in \mathcal{D}$, $j = 1, 2, \dots, p$ denotes the output node, and $f_i^{\Sigma(t)} : \mathcal{D}^n \rightarrow \mathcal{D}$, $h_j^{\Sigma(t)} : \mathcal{D}^n \rightarrow \mathcal{D}$ are Boolean functions. In what follows we use $Y(t) := \{Y_1(t), Y_2(t), \dots, Y_p(t)\}$.

Definition 3.1. SBN (3.1) is said to be identifiable, if f_i^k and h_j^k , $i = 1, 2, \dots, n$, $j = 1, 2, \dots, p$, $k = 1, 2, \dots, \mu$, can be uniquely determined via certain switching signal and observed output data $\{\Sigma(0), \Sigma(1), \dots\}$ and $\{Y(0), Y(1), \dots\}$.

Relying on Lemma 2.4, the equivalent algebraic form of system (3.1) is:

$$\begin{cases} x(t+1) = L_{\Sigma(t)}x(t), \\ y(t) = H_{\Sigma(t)}x(t), \end{cases} \quad (3.2)$$

where $x(t) = \bowtie_{i=1}^n x_i(t) \in \Delta_{2^n}$, $y(t) = \bowtie_{j=1}^p y_j(t) \in \Delta_{2^p}$, and $L_k \in \mathcal{L}_{2^n \times 2^n}$ and $H_k \in \mathcal{L}_{2^p \times 2^n}$ are the state and output structure matrices of the k -th subnetwork, respectively.

Set $L = [L_1, L_2, \dots, L_\mu] \in \mathcal{L}_{2^n \times \mu \cdot 2^n}$ and $H = [H_1, H_2, \dots, H_\mu] \in \mathcal{L}_{2^p \times \mu \cdot 2^n}$. Let $\Sigma \sim \sigma$, where $\sigma(t) \in \Delta_\mu$. Then, SBN (3.2) is equivalent to

$$\begin{cases} x(t+1) = L\sigma(t)x(t), \\ y(t) = H\sigma(t)x(t), \end{cases} \quad (3.3)$$

Obviously, $L\sigma(t) = L_{\Sigma(t)}$ and $H\sigma(t) = H_{\Sigma(t)}$.

From the algebraic formulations presented in (3.2) and (3.3), it is evident that identifying the matrices L and H is equivalent to identifying the structure matrices of each subnetwork L_k and H_k with $k = 1, 2, \dots, \mu$. Furthermore, as outlined in Proposition 7.3 of [30], we can deduce the Boolean functions of each node in the k -th subnetwork, i.e., f_i^k and h_j^k , from the matrices L_k and H_k . This can be done by following the procedure for reconstructing the original system through its structure matrices. Consequently, the identification problem stated in Definition 3.1 can be

equivalently redefined as: *SBN (3.3) is said to be identifiable if each column of matrices L and H can be determined from certain switching signal and observed output data $\{\sigma(0), \sigma(1), \dots\}$ and $\{y(0), y(1), \dots\}$.*

Consider a coordinate transformation $x'(t) = Mx(t)$ with $M \in \mathcal{L}_{2^n \times 2^n}$; applying this transformation to system (3.3), we obtain:

$$\begin{cases} x'(t+1) = L'\sigma(t)x'(t), \\ y(t) = H'\sigma(t)x'(t), \end{cases} \quad (3.4)$$

where L' and H' are defined as $ML(I_\mu \otimes M^T)$ and $H(I_\mu \otimes M^T)$ respectively. Clearly, the tuple (L, H) of system (3.3) and (L', H') of system (3.4) are not distinguishable by any switching-output data. Thus, identification leads to identifying an equivalence class of (L, H) , rather than a specific (L, H) . Keeping this in mind, an identifiable SBN is also said to be uniquely identifiable.

System (3.1) (and equivalently system (3.3)) consists of μ subnetworks, each with its unique output logical function, which differs it from traditional SBNS, where output functions are typically identical, as seen in [8, 66]. For traditional SBNS, identification can be addressed by methods in [51, 54], considering switching signals as inputs. However, these methods do not apply to system (3.1); the aim of our research is to provide such a method.

Assumption 3.2. ([54]) *The available switching signal and observed data are sufficient to describe all possible switching-output pairs in the system.*

Remark 3.3. *The assumption is reasonable since it occurs in the real world. For example, when we study the Drosophila segment polarity network, the samples we need to collect are abundant, which means that we maybe select a lot of data from different drosophila.*

In this chapter, we consider two possible scenarios. In the first, the single-sample scenario, data are obtained from a fixed initial state. The second scenario, called the multiple-sample scenario, corresponds to the case where samples are derived from diverse initial states. Based on these two scenarios, we pose the following two identification problems.

Problem 1: *Given q groups of switching-output data $\{(\sigma^i(0), y^i(0)), (\sigma^i(1), y^i(1)), \dots, (\sigma^i(r), y^i(r))\}$, where $i \in \{1, 2, \dots, q\}$ and $r \in \mathbb{Z}_+$, all sampled from the same initial state, find the matrices L and H of SBN (3.3).*

Problem 2: Given mq groups of switching-output data $\{(\sigma^{j,i}(0), y^{j,i}(0)), (\sigma^{j,i}(1), y^{j,i}(1)), \dots, (\sigma^{j,i}(r), y^{j,i}(r)))\}$, where $j \in \{1, 2, \dots, m\}$, $i \in \{1, 2, \dots, q\}$ and $r \in \mathbb{Z}_+$, all sampled from m distinct initial states, find the matrices L and H of SBN (3.3).

To facilitate a clearer understanding of these two scenarios, we provide more detailed explanations. When studying the *Drosophila* segment polarity network, data from numerous samples are collected. Research indicates that segment polarity genes enhance and maintain their expression through intricate intracellular and intercellular regulatory interactions [128]. The segment formation during *Drosophila* embryonic development involves extensive gene expression and regulation. Problem 1 involves data collection from q *drosophila* specimens in the same testing environment, where their initial states are identical, i.e., $x^1(0) = x^2(0) = \dots = x^q(0)$. In contrast, Problem 2 involves data collection from q *drosophila* specimens across m different testing environments. Here, these mq data groups originate from m distinct initial states, such that for each $j \in \{1, 2, \dots, m\}$, $x^{j,1}(0) = x^{j,2}(0) = \dots = x^{j,q}(0)$. It is important to note that the length of the collected data may vary, which could lead to different values for r in each switching-output sequence.

3.2 Reachability and Observability of SBNS

Reference [51] demonstrated that a BCN is identifiable from input-output data with suitable controls if and only if it is both controllable and observable. Additionally, [54] proved that a BN is identifiable under a multiple-sample case if and only if it is observable, among other conditions. Motivated by these findings, this section will discuss reachability and observability of SBNS.

First, we provide the definition of reachability of SBN (3.3).

Definition 3.4. Consider SBN (3.3) and let $x(0) := x_0 \in \Delta_{2^n}$.

- 1) (Point switching reachability) A state $x \in \Delta_{2^n}$ is said to be point switching reachable from x_0 , if there exists a time instant $s \in \mathbb{N}$ and a switching signal sequence $\{\sigma(0), \sigma(1), \dots, \sigma(s-1)\}$, such that under the switching signal sequence, the trajectory of SBN (3.3) starting from x_0 reaches x at time s ;
 - 2) (Switching reachability) SBN (3.3) is said to be switching reachable from x_0 , if for any state $x \in \Delta_{2^n}$, there exists a time instant $s \in \mathbb{N}$ and a switching signal sequence $\{\sigma(0), \sigma(1), \dots, \sigma(s-1)\}$, such that when SBN (3.3) is initialized in x_0 and the switching sequence is applied, $x(s) = x$.
-

- 3) (Global switching reachability) SBN (3.3) is said to be globally switching reachable, if for any initial state $x_0 \in \Delta_{2^n}$, SBN (3.3) is switching reachable from x_0 .

Note that the point switching reachability in Definition 3.4 is adapted from [9], the last two definitions are the generalizations of the first one. Next we give the conditions to detect the reachability of SBNS (3.3). We introduce the following lemma.

Lemma 3.5. ([9]) Consider SBN (3.3) and let $A := \Sigma_{i=1}^{\mu} L_i$. State $x = \delta_{2^n}^b$ is point switching reachable from $x_0 = \delta_{2^n}^a$, if and only if

$$(\Sigma_{j=1}^{2^n} A^{(j)})_{b,a} > 0.$$

From the above lemma, if any state $\delta_{2^n}^b \in \Delta_{2^n}$ is point switching reachable from $x_0 = \delta_{2^n}^a$, then $(\Sigma_{j=1}^{2^n} A^{(j)})_{b,a} > 0$ holds for any $b \in \{1, 2, \dots, 2^n\}$, i.e., $\text{Col}_a(\Sigma_{j=1}^{2^n} A^{(j)}) > \mathbf{0}_{2^n}$. Moreover, if any state $\delta_{2^n}^b \in \Delta_{2^n}$ can be reached from any $x_0 = \delta_{2^n}^a \in \Delta_{2^n}$, then $(\Sigma_{j=1}^{2^n} A^{(j)})_{b,a} > 0$ holds for any $a, b \in \{1, 2, \dots, 2^n\}$, i.e., $\Sigma_{j=1}^{2^n} A^{(j)} > \mathbf{0}_{2^n \times 2^n}$. These considerations lead to the conditions for the reachability of SBN (3.3) as reported in the following corollary.

Corollary 3.6. Consider SBN (3.3) and let $A := \Sigma_{i=1}^{\mu} L_i$. Then

- 1) SBN (3.3) is switching reachable from $x_0 = \delta_{2^n}^a \in \Delta_{2^n}$, if and only if

$$\text{Col}_a(\Sigma_{j=1}^{2^n} A^{(j)}) > \mathbf{0}_{2^n};$$

- 2) SBN (3.3) is globally switching reachable, if and only if

$$\Sigma_{j=1}^{2^n} A^{(j)} > \mathbf{0}_{2^n \times 2^n}.$$

The observability of general SBNS (i.e., where both state and output are influenced by switching signals) with control inputs was proposed in [12]; leveraging on that, we define the observability of SBNS as follows:

Definition 3.7. SBN (3.3) is said to be observable if there exists a switching signal sequence $\{\sigma(0), \sigma(1), \dots, \sigma(l)\}$, where $l \in \mathbb{Z}_+$, such that for any two different initial states x_0 and $\bar{x}_0 \in \Delta_{2^n}$, the corresponding output trajectories satisfy $\{y(0), y(1), \dots, y(l)\} \neq \{\bar{y}(0), \bar{y}(1), \dots, \bar{y}(l)\}$.

If for two different states $x_0, \bar{x}_0 \in \Delta_{2^n}$, the output $y(l) = \bar{y}(l)$ for any $l \in \mathbb{N}$ when any switching sequence is applied, the pair of states $(x_0, \bar{x}_0)$ are called indistinguishable; otherwise, they are distinguishable. We here remark that in [12], the observability is investigated using the deterministic finite automaton method. In what follows we propose a matrix-based method, independent of deterministic finite automaton knowledge, to check the observability of SBN (3.3).

Based on Definition 3.7, if two different initial states $x_0 = \delta_{2^n}^a$ and $\bar{x}_0 = \delta_{2^n}^b$ yield identical outputs in the i -th subnetwork, i.e., $y(0) = H_i \delta_{2^n}^a = H_i \delta_{2^n}^b = \bar{y}(0)$, this implies

$$1 = (H_i \delta_{2^n}^a)^T H_i \delta_{2^n}^a = (H_i \delta_{2^n}^a)^T H_i \delta_{2^n}^b = (\delta_{2^n}^a)^T H_i^T H_i \delta_{2^n}^b = (H_i^T H_i)_{a,b}.$$

Since H_i is a logical matrix, each element of $H_i^T H_i$ is either 1 or 0. $(H_i^T H_i)_{a,b} = 1$ implies that $\delta_{2^n}^a$ and $\delta_{2^n}^b$ are indistinguishable in the i -th subnetwork at $t = 0$. Conversely, if $(H_i^T H_i)_{a,b} = 0$, i.e., $(H_i \delta_{2^n}^a)^T H_i \delta_{2^n}^b = 0$, then $\delta_{2^n}^a$ and $\delta_{2^n}^b$ are distinguishable in the i -th subnetwork. Clearly, $(H_i^T H_i)_{a,a} = 1$ always holds, as state $\delta_{2^n}^a$ is indistinguishable from itself. Consequently, $H_i^T H_i = I_{2^n}$ implies that at $t = 0$, any pair of different initial states are distinguishable in the i -th subnetwork.

Construct two sets of matrices $\mathcal{Q}^0$ and $\mathcal{G}^0$ as

$$\mathcal{Q}^0 = \mathcal{G}^0 := \{Q_i^0 = G_i^0 := H_i^T H_i \mid i \in \{1, 2, \dots, \mu\}\}. \quad (3.5)$$

Evidently, $I_{2^n} \in \mathcal{Q}^0 (\mathcal{G}^0)$ indicates that there exists at least an integer $j \in \{1, 2, \dots, \mu\}$ such that $I_{2^n} = Q_j^0 (G_j^0)$. This implies that there exists at least a switching signal $\sigma(0) = \delta_\mu^j$ such that any pair of different initial states are distinguishable.

Further, we consider

$$\begin{cases} x(1) = L\sigma(0)x(0), \\ y(1) = H\sigma(1)x(1) = H\sigma(1)L\sigma(0)x(0). \end{cases}$$

Given two different initial states $\delta_{2^n}^a$ and $\delta_{2^n}^b$, the switching signals $\sigma(0) = \delta_\mu^{i_0}$, $i_0 \in \{1, 2, \dots, \mu\}$ and $\sigma(1) = \delta_\mu^{i_1}$, $i_1 \in \{1, 2, \dots, \mu\}$, and the corresponding outputs $y(1)$ and $\bar{y}(1)$. If it holds $y(1) = \bar{y}(1)$, then

$$((H_{i_1} L_{i_0})^T H_{i_1} L_{i_0})_{a,b} = (L_{i_0}^T H_{i_1}^T H_{i_1} L_{i_0})_{a,b} = (L_{i_0}^T (H_{i_1}^T H_{i_1}) L_{i_0})_{a,b} = (L_{i_0}^T Q_{i_1}^0 L_{i_0})_{a,b} = 1.$$

When considering all possible switching signals $\sigma(0)$ and $\sigma(1)$, we construct two

novel sets of matrices $\mathcal{Q}^1$ and $\mathcal{G}^1$ defined as follows:

$$\mathcal{Q}^1 := \{Q_{(k-1)\mu+i}^1 := L_k^T Q_i^0 L_k \mid Q_i^0 \in \mathcal{Q}^0, i \in \{1, 2, \dots, \mu\}, k \in \{1, 2, \dots, \mu\}\} \quad (3.6)$$

and

$$\mathcal{G}^1 := \{G_j^1 := Q_j^1 \wedge G_{\lceil (j-1)/\mu \rceil}^0 \mid Q_j^1 \in \mathcal{Q}^1, G_{\lceil (j-1)/\mu \rceil}^0 \in \mathcal{G}^0, j \in \{1, 2, \dots, \mu^2\}\}. \quad (3.7)$$

Note that $j \in \{1, 2, \dots, \mu^2\}$ since the maximum value of $(k-1)\mu + i$ equals μ^2 when $k = \mu$ and $i = \mu$.

From the j -th matrix $Q_{(j_0-1)\mu+j_1}^1$ in set $\mathcal{Q}^1$, where $j := (j_0 - 1)\mu + j_1$, we can determine whether under the switching sequence $\{\sigma(0), \sigma(1)\} = \{\delta_\mu^{j_0}, \delta_\mu^{j_1}\} := \delta_\mu\{j_0, j_1\}$ the outputs at time $t = 1$ starting from any distinct initial state are the same. Thus, the set $\mathcal{Q}^1$ is constructed to check whether for any pair of distinct initial state, the corresponding outputs of SBN (3.3) at time $t = 1$ are the same under any switching sequence $\{\sigma(0), \sigma(1)\}$. While $\mathcal{G}^1$, combined with elements in $\mathcal{G}^0$, encompasses all possible information to detect whether SBN (3.3) is observable within one step. Note that when constructing matrix G_j^1 , the elements from $\mathcal{G}^0$ and $\mathcal{Q}^0$ cannot be arbitrarily combined. Instead, the element in $\mathcal{Q}^0$ under a specific switching sequence $\{\sigma(0), \sigma(1)\}$ should be combined only with the element in $\mathcal{G}^0$ associated with the switching signal $\sigma(0)$, which guarantees that the constructed set $\mathcal{G}^1$ accurately captures the system's observability properties within one step. Specifically, $I_{2^n} \in \mathcal{G}^1$ implies the existence of an integer $j = (j_0 - 1)\mu + j_1 \in \{1, 2, \dots, \mu^2\}$ such that

$$I_{2^n} = G_j^1 = Q_j^1 \wedge G_{j_0}^0 = (L_{j_0}^T Q_{j_1}^0 L_{j_0}) \wedge G_{j_0}^0 = (L_{j_0}^T H_{j_1}^T H_{j_1} L_{j_0}) \wedge (H_{j_0}^T H_{j_0}).$$

This ensures that a specific switching sequence $\{\sigma(0), \sigma(1)\} = \delta_\mu\{j_0, j_1\}$ exists such that for any two distinct initial states $\delta_{2^n}^a \neq \delta_{2^n}^b$, distinct outputs are generated, fulfilling

$$\{y(0), y(1)\} = \{H_{j_0} \delta_{2^n}^a, H_{j_1} L_{j_0} \delta_{2^n}^a\} \neq \{H_{j_0} \delta_{2^n}^b, H_{j_1} L_{j_0} \delta_{2^n}^b\} = \{\bar{y}(0), \bar{y}(1)\}.$$

Iterating this process, we can generate the matrix sets $\mathcal{Q}^l$ and $\mathcal{G}^l$ for $l > 1$, defined as:

$$\mathcal{Q}^l := \{Q_{(k-1)\mu^l+i}^l = L_k^T Q_i^{l-1} L_k \mid Q_i^{l-1} \in \mathcal{Q}^{l-1}, i \in \{1, 2, \dots, \mu^l\}, k \in \{1, 2, \dots, \mu\}\} \quad (3.8)$$

and

$$\mathcal{G}^l := \{G_j^l = Q_j^l \wedge G_{[(j-1)/\mu]}^{l-1} \mid Q_j^l \in \mathcal{Q}^l, G_{[(j-1)/\mu]}^{l-1} \in \mathcal{G}^{l-1}\}, \quad (3.9)$$

$$j \in \{1, 2, \dots, \mu^{l+1}\}.$$

If there exists $G_j^l \in \mathcal{G}$ satisfying $I_{2^n} = G_j^l \in \mathcal{G}^l$, then there exists a switching sequence $\{\sigma(0), \sigma(1), \dots, \sigma(l)\}$ such that any pair of distinct initial states $(\delta_{2^n}^a, \delta_{2^n}^b)$ are distinguishable. For any $G_j^l \in \mathcal{G}^l$ and for any $j \in \{1, 2, \dots, \mu^{l+1}\}$, we can identify unique integers $j_0, j_1, \dots, j_l \in \{1, 2, \dots, \mu\}$ such that $\delta_{\mu^{l+1}}^j$ can be written as

$$\delta_{\mu^{l+1}}^j = \delta_{\mu}^{j_0} \delta_{\mu}^{j_1} \dots \delta_{\mu}^{j_l}. \quad (3.10)$$

Denote $\delta_{\mu^l}^{\bar{j}} := \delta_{\mu}^{j_1} \delta_{\mu}^{j_2} \dots \delta_{\mu}^{j_l}$ and $\delta_{\mu^l}^{\tilde{j}} := \delta_{\mu}^{j_0} \delta_{\mu}^{j_1} \dots \delta_{\mu}^{j_{l-1}}$. Then, $\delta_{\mu^{l+1}}^j = \delta_{\mu}^{j_0} \delta_{\mu^l}^{\bar{j}} = \delta_{\mu^l}^{\tilde{j}} \delta_{\mu}^{j_l}$. This implies

$$G_j^l = Q_j^l \wedge G_{\tilde{j}}^{l-1} = (L_{j_0}^T Q_{\tilde{j}}^{l-1} L_{j_0}) \wedge G_{\tilde{j}}^{l-1}.$$

Hence, if $G_j^l = I_{2^n}$, there exists a switching sequence $\{\sigma(0), \sigma(1), \dots, \sigma(l)\} = \delta_{\mu}\{j_0, j_1, \dots, j_l\}$ such that every pair of initial states $(\delta_{2^n}^a, \delta_{2^n}^b)$ with $\delta_{2^n}^a \neq \delta_{2^n}^b$ are distinguishable, this implies that

$$\{y(0), y(1), \dots, y(j)\} \neq \{\bar{y}(0), \bar{y}(1), \dots, \bar{y}(j)\},$$

where $y(0) = H_{j_0} \delta_{2^n}^a$, $\bar{y}(0) = H_{j_0} \delta_{2^n}^b$, and $y(r) = H_{j_r} L_{j_{r-1}} \dots L_{j_0} \delta_{2^n}^a$, $\bar{y}(r) = H_{j_r} L_{j_{r-1}} \dots L_{j_0} \delta_{2^n}^b$, $r \in \{1, 2, \dots, l\}$.

Finally, if the outputs of all states in each subnetwork are the same, SBN (3.3) will never be observable. Thus, to make the observability problem meaningful we make the following assumption.

Assumption 3.8. Given SBN (3.3), $(H_i^T H_i) \neq \mathbf{1}_{2^n \times 2^n}$ for at least one $i \in \{1, 2, \dots, \mu\}$.

In what follows, we assume that SBN (3.3) verifies Assumption 3.8. Based on the above analysis, we derive a direct criterion for the observability of SBNS.

Theorem 3.9. SBN (3.3) is observable if and only if there exists an integer $0 \leq l^* \leq \hat{l}$, such that

$$I_{2^n} \in \mathcal{G}^{l^*}.$$

Remark 3.10. Since in an SBN, the number of the states and switching signals are given, $\hat{l}$ is finite. Its value depends on the structure of subnetworks. The worst case is given by all state are collected in all subnetworks and given by the switching signals. Thus, the worst case for $\hat{l}$ is $\hat{l} = (2^{2^n-1} - 2^{n-1})\mu$.

Moreover, we provide the following algorithm to determine the observability of SBN (3.3).

Algorithm 1 Determining whether SBN (3.3) is observable

Require: Matrices L and H , and the integer $\hat{l}$

Ensure: SBN (3.3) is (not) observable

```

1: Initialization: Set  $l \leftarrow 1$ ,  $l^* \leftarrow 0$ .
2: Construct two sets of matrices  $\mathcal{Q}^0$  and  $\mathcal{G}^0$  according to (3.5).
3: if  $I_{2^n} \in \mathcal{G}^0$  then
4:   return: SBN (3.3) is observable,  $l^* = 0$ .
5: else
6:   while  $l \leq \hat{l}$  do
7:     Construct two sets of matrices  $\mathcal{Q}^l$  and  $\mathcal{G}^l$  according to (3.6) and (3.7).
8:     if  $I_{2^n} \in \mathcal{G}^l$  then
9:        $l^* \leftarrow l$ 
10:      return: SBN (3.3) is observable,  $l^* = l$ .
11:     else
12:        $l \leftarrow l + 1$ 
13:     end if
14:   end while
15:   if  $l > \hat{l}$  then
16:     return: SBN (3.3) is not observable.
17:   end if
18: end if

```

Remark 3.11. The method presented in Algorithm 1 only relies on structure matrices of the system (3.3) and can be easily implemented in MATLAB. Thus, compared with the approach in [12], which requires knowledge of the deterministic finite automaton, the method proposed here is more straightforward. Compared to the set reachability method under parallel systems in [66], we compute matrices with lower dimensions, that is, $2^n \times 2^n$ rather than $2^{2n} \times 2^{2n}$.

Note that for any matrix in $\mathcal{G}^{l^*}$ that is equal to I_{2^n} , we are able to reconstruct the switching sequence such that SBN (3.3) is observable. This can be done directly by applying equation (3.10). We denote such sequence as $\{\sigma^*(0), \sigma^*(1), \dots, \sigma^*(l^*)\}$ and refer to it as the O-switching sequence.

Definition 3.12. Given an SBN (3.3), if there exists integers $l^* \in N$ and $j \in \{1, 2, \dots, \mu^{l^*}\}$, such that $G_j^{l^*} = I_{2^n}$ holds, then the switching signal sequence $\{\sigma^*(0), \sigma^*(1), \dots, \sigma^*(l^*)\} = \delta_\mu\{j_0, j_1, \dots, j_{l^*}\}$, where $j_i = \lceil j / \mu^{l^*-i} \rceil - (\lceil j / \mu^{l^*-i+1} \rceil - 1)\mu$ with $i \in \{0, 1, \dots, l^*\}$, is an O-switching sequence.

Example 3.13. Given an SBN with 2 subnetworks, 3 state nodes and 1 output as

$$\begin{cases} x(t+1) = L_i x(t), \\ y(t) = H_i x(t), \end{cases} \quad (3.11)$$

where $i = 1, 2$, $L_1 = \delta_8[2, 1, 5, 3, 8, 7, 6, 4]$, $L_2 = \delta_8[1, 8, 5, 4, 3, 4, 7, 8]$, $H_1 = \delta_2[1, 2, 2, 1, 2, 2, 1, 1]$ and $H_2 = \delta_2[1, 2, 1, 1, 1, 2, 2, 1]$.

We first calculate

$$H_1^T H_1 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad H_2^T H_2 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

According to (3.5), we have

$$\mathcal{Q}^0 = \mathcal{G}^0 := \{H_1^T H_1, H_2^T H_2\}.$$

It is easy to show $I_{8 \times 8} \notin \mathcal{Q}^0$ ($\mathcal{G}^0$). We then calculate $\mathcal{Q}^l$ and $\mathcal{G}^l$ according to (3.8) and (3.9) until $l = 3$. In set $\mathcal{G}^3$, we can find $G_4^3 = G_5^3 = G_7^3 = G_9^3 = G_{11}^3 = I_{8 \times 8}$, which implies that the system (3.11) is observable.

If considered separately, both subnetwork 1 and subnetwork 2 are not observable according to [64]. From Figure 3.1, we can intuitively see that states δ_8^2 and δ_8^6 are indistinguishable in both the first and second subnetwork. However, when two subnetworks are connected by switching signals, the generated SBN (3.11) is observable.

Additionally, 5 O-switching sequences can be straightly obtained. Based on (3.10), we get that $\delta_{16}^4 = \delta_2^1 \delta_2^1 \delta_2^2 \delta_2^2$. Therefore, SBN (3.11) is observable under the O-switching sequence $\{\sigma^*(0), \sigma^*(1), \sigma^*(2), \sigma^*(3)\} = \delta_2\{1, 1, 2, 2\}$. The other 4 O-switching sequences can be obtained similarly and are omitted here.

Based on the reachability and observability of SBNs, we examine the conditions necessary for identifying SBNs and address Problems 1 and 2 in the next two sections. Although the switching signal can be viewed as a control input, it is crucial to note that the SBN considered in this chapter comprises multiple distinct BNs,

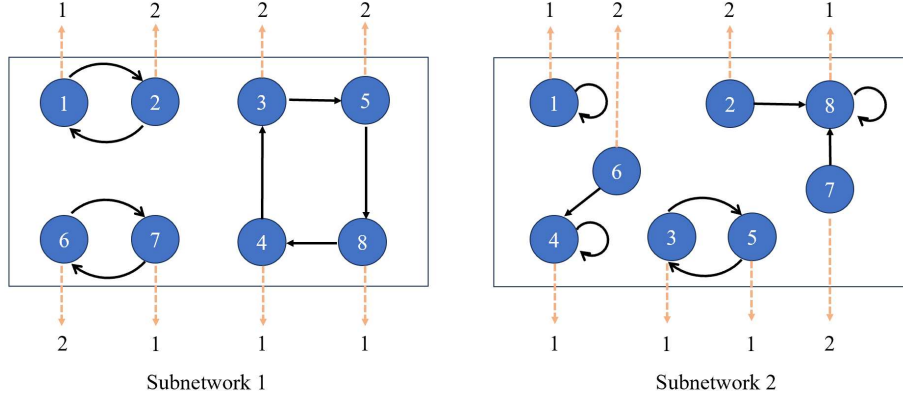


Figure 3.1: The state transition diagram of system (3.11), where blue balls stand for states and yellow dashed arrows point to the outputs of the states.

differing from the BCN discussed in [54]. The complexity of identifying SBNS (3.3) is heightened by the variability in outputs across each subnetwork, as the switching signals influence not only the states but also the outputs.

3.3 Identification of SBNS under a Single-sample Scenario

The property of global switching reachability given in Definition 3.1 allows the identification of the ‘evolutionary process’ of all states. It indicates that there exists a switching signal sequence $\{\sigma(0), \sigma(1), \dots, \sigma(s)\}$ such that the switching-state pair $(\sigma(t), x(t))$ for $t \in \{0, 1, 2, \dots, s\}$ covers all possible pairs $(\delta_\mu^i, \delta_{2^n}^j)$, where $i \in \{0, 1, \dots, \mu\}$, and $j \in \{0, 1, \dots, 2^n\}$. The observability property ensures the identification of 2^n distinct initial states (i.e. the entire state space) from 2^n different output sequences. Then, leveraging these two properties, we address Problem 1 of SBN (3.3).

For an observable and globally reachable SBN (3.3), let $\{\sigma^*(0), \sigma^*(1), \dots, \sigma^*(l^*)\}$ represent an O-switching sequence, and let $\{\sigma(0), \sigma(1), \dots, \sigma(s)\}$ with $s \geq \mu \cdot 2^n$ be termed an R-switching sequence satisfying

$$\{(\sigma(t), x(t)) \mid t \in \{0, 1, \dots, s\}\} = \Delta_{\mu \cdot 2^n}. \quad (3.12)$$

From the above switching sequences, we construct the following series of switching

sequences as the given switching signal data:

$$\begin{aligned}
S^0(l^*) &:= \{\sigma^*(0), \sigma^*(1), \dots, \sigma^*(l^*)\}, \\
S^1(l^* + 1) &:= \{\sigma(0), \sigma^*(0), \sigma^*(1), \dots, \sigma^*(l^*)\}, \\
S^2(l^* + 2) &:= \{\sigma(0), \sigma(1), \sigma^*(0), \sigma^*(1), \dots, \sigma^*(l^*)\}, \\
&\vdots \\
S^{s+1}(l^* + s + 1) &:= \{\sigma(0), \sigma(1), \dots, \sigma(s), \sigma^*(0), \sigma^*(1), \dots, \sigma^*(l^*)\}.
\end{aligned} \tag{3.13}$$

Furthermore, given the initial state $x^0(0) = x^1(0) = \dots = x^{s+1}(0)$, by applying the switching sequences (3.13), we obtain the corresponding observed data:

$$\begin{aligned}
P^0(l^*) &:= \{y^0(0), y^0(1), y^0(2), \dots, y^0(l^*)\}, \\
P^1(l^* + 1) &:= \{y^1(0), y^1(1), y^1(2), \dots, y^1(l^* + 1)\}, \\
P^2(l^* + 2) &:= \{y^2(0), y^2(1), y^2(2), \dots, y^2(l^* + 2)\}, \\
&\vdots \\
P^{s+1}(l^* + s + 1) &:= \{y^{s+1}(0), y^{s+1}(1), y^{s+1}(2), \dots, y^{s+1}(l^* + s + 1)\}.
\end{aligned} \tag{3.14}$$

Clearly, $P^i(l^* + i)$ is the output sequence of SBN (3.3) starting from $x^i(0)$ under the switching sequence $S^i(l^* + i)$, where $i \in \{1, 2, \dots, s + 1\}$. In the following, we propose a condition for the identification of SBN (3.3).

Theorem 3.14. *SBN (3.3) is uniquely identifiable under a single-sample scenario if it is observable and globally switching reachable.*

Proof: Assuming that SBN (3.3) is both observable and globally switching reachable, we can derive its R-switching sequence $\{\sigma(0), \sigma(1), \dots, \sigma(s)\}$ satisfying equation (3.12) as well as an O-switching sequence $\{\sigma^*(0), \sigma^*(1), \dots, \sigma^*(l^*)\}$. This allows us to derive equation (3.13) and obtain equation (3.14). Considering the output data (3.14), we denote by $\bar{P}^i$ the output sequence consisting of the last $l^* + 1$ elements of $P^i(l^* + i)$, where $i \in \{0, 1, \dots, s + 1\}$.

Due to the observability property, there must be 2^n distinct output sequences in all data $\bar{P}^i$ with $i \in \{0, 1, \dots, s + 1\}$. They can be denoted as $Y^1, Y^2, \dots, Y^{2^n}$ and can be determined according to the order in which a new output sequence appears while scanning all $\bar{P}^i$ with $i \in \{0, 1, \dots, s + 1\}$. Clearly, $Y^1 = \bar{P}^0$. Treating Y^j as the output sequences of state $\delta_{2^n}^j$ and letting $\tilde{x}(i) = \delta_{2^n}^j$ when $\bar{P}^i = Y^j$, we can obtain the final state trajectory $\tilde{x}(0), \tilde{x}(1), \dots, \tilde{x}(s + 1)$. Concurrently, we rewrite the element $y^{i+1}(i)$ in each data $P^{i+1}(l^* + i + 1)$ by $\tilde{y}(i)$, where $i \in \{0, 1, \dots, s\}$.

With the R-switching sequence $\{\sigma(0), \sigma(1), \dots, \sigma(s)\}$, the ‘evolutionary order’ of these 2^n states and corresponding outputs in each subnetwork can be determined.

According to

$$\begin{cases} \tilde{x}(i+1) = L\sigma(i)\tilde{x}(i), \\ \tilde{y}(i) = H\sigma(i)\tilde{x}(i), \end{cases} \quad (3.15)$$

where $i \in \{0, 1, \dots, s\}$, we can infer the structure matrices L and H . Therefore, SBN (3.3) is identifiable.

We turn the analysis of Theorem 3.14 into an algorithmic procedure with steps given in Algorithm 2.

Algorithm 2 Identify an observable and globally switching reachable SBN (3.3) under a single-sample scenario

Require: $s+2$ groups of switching signal data (3.13)

Ensure: Matrices L and H

- 1: Generate $s+2$ groups of output data (3.14) and denote the output sequence consisting of the last l^*+1 elements of $P^i(i+l^*)$ by $\bar{P}^i$, where $i \in \{0, 1, \dots, s+1\}$.
 - 2: **Initialization:** Set $j \leftarrow 1$, $\mathcal{Y} \leftarrow \emptyset$, $\tilde{x}(0) \leftarrow \delta_{2^n}^1$, $\tilde{y}(0) \leftarrow y^1(0)$.
 - 3: **for** $i = 0$ **to** $s+1$ **do**
 - 4: **if** $\bar{P}^i \notin \mathcal{Y}$ **then**
 - 5: $Y^j = \bar{P}^i$, $\mathcal{Y} = \mathcal{Y} \cup \{Y^j\}$, $\tilde{x}(i) = \delta_{2^n}^j$, and $j \leftarrow j+1$
 - 6: **else**
 - 7: **for** $k = 1$ **to** $j-1$ **do**
 - 8: **if** $\bar{P}^i = Y^k$ **then**
 - 9: $\tilde{x}(i) = \delta_{2^n}^k$
 - 10: **end if**
 - 11: **end for**
 - 12: **end if**
 - 13: $\tilde{y}(i) = y^{i+1}(i)$
 - 14: **end for**
 - 15: Construct L and H according to equations (3.15).
-

Figure 3.2 illustrates the identification process of an observable and globally switching reachable SBN under a single-sample scenario. There are many groups of switching-output data, and we only list the first 3 groups of data, stemming from a given initial state. Clearly, $x^0(0) = x^1(0) = x^2(0)$. The first step is to identify some initial states of the system, denoted by $\tilde{x}(0)$, $\tilde{x}(1)$, and $\tilde{x}(2)$, by scanning output sequences $\bar{P}^0$, $\bar{P}^1$ and $\bar{P}^2$, shown in the dashed-line rectangle box. According to the property of observability, one-to-one correspondence, $\bar{P}^0$ can be interpreted as the output sequence of state $\delta_{2^n}^1$ and then we identify $\tilde{x}(0) = \delta_{2^n}^1$. In the second group of data, if $\bar{P}^1 \neq \bar{P}^0$, we identify $\bar{P}^1$ as the output sequence of state $\delta_{2^n}^2$, and

$\tilde{x}(1) = \delta_{2n}^2$ can be correspondingly identified. Otherwise, $\tilde{x}(1) = \delta_{2n}^1$. Similarly, we can identify $\tilde{x}(2)$ according to $\bar{P}^2$. Therefore, we obtain the corresponding state sequence $\{\tilde{x}(0), \tilde{x}(1), \tilde{x}(2)\}$ under the switching sequence $\{\sigma(0), \sigma(1)\}$.

The second step is to identify the corresponding outputs. Since switching signals affect both input and output consistently, we need to focus on the output data generated under $\{\sigma(0), \sigma(1)\}$. Thus, the relevant output sequence $\{\tilde{y}(0), \tilde{y}(1)\}$ can be identified as $\{y^1(0), y^2(1)\}$ via the relation in the solid-line rectangle box. The process continues until all data are collected, allowing us to identify complete the state and output sequences and further identify all columns of L and H since the system is globally switching reachable.

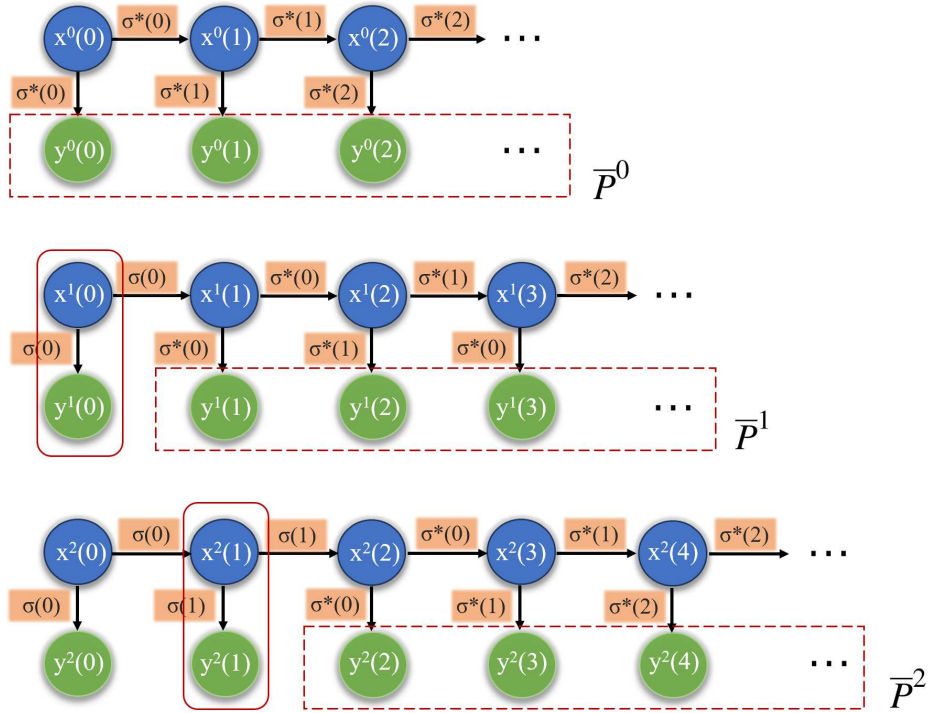


Figure 3.2: The partial identification process of SBN (3.3), where blue balls, green balls and yellow rectangles stand for states, outputs data and switching signal data, respectively.

Remark 3.15. Algorithm 3.2 in [54] presented a method for identifying BCNs. However, if we regard the switching signals as controls and apply the same method to identify SBN (3.3), we can only determine L . Attempting to identify H using this method will yield incorrect results. This discrepancy arises because selecting the first element of Y^i as the output of state δ_{2n}^i allows us to identify only $H_{\sigma^*(0)}$ under the switching signal $\sigma^*(0)$. However, our algorithm finds the exact correspon-

dence between states and outputs by taking time into account and then enables the identification of the entire matrix H .

Furthermore, in the single-sample scenario, we can formulate a necessary and sufficient condition for identifying SBN (3.3). To this end, SBN (3.3) must satisfy *Stronger switching reachability*, that is, there exists a time instant $s \in \mathbb{N}$ and a switching signal sequence $\{\sigma(0), \sigma(1), \dots, \sigma(s-1)\}$, such that $x(0) = x_0$ and for all $x \in \Delta_{2^n}$, there exists an $i \in \{0, 1, \dots, s\}$ such that $x(i) = x$.

Corollary 3.16. *Starting from a fixed initial state x_0 , SBN (3.3) is uniquely identifiable under a single-sample scenario if and only if it is observable and stronger switching reachable from x_0 .*

Proof: (Sufficiency) System (3.3) is stronger switching reachable from x_0 indicates that there exists a switching sequence satisfying (3.12) with x_0 . Similar to the proof of Theorem 3.14, combined with the observability property, we infer that SBN (3.3) is identifiable.

(Necessity) It is proved by contradiction. Suppose the system is not observable. In that case, there exist at least two initial states $x_0 \neq \bar{x}_0$, such that their output sequences are identical under arbitrary switching sequence. This implies that we can only identify less than 2^n distinct output sequences from all sequences $\bar{P}^i$, $i \in \{0, 1, \dots, s+1\}$, indicating that a one-to-one correspondence between Y^j and $\delta_{2^n}^j$ cannot be obtained. Similarly, if the system is not strongly switching reachable from x_0 , we cannot find any switching sequence satisfying (3.12) with $x(0) = x_0$, and consequently, we cannot fully determine the ‘evolutionary rule’ of the obtained state trajectory $\{\tilde{x}(0), \tilde{x}(1), \dots, \tilde{x}(s)\}$ under all subnetworks. Both scenarios contradict the fact that the system is identifiable. Therefore, SBN (3.3) is observable and stronger switching reachable from x_0 . $\square$

3.4 Identification of SBNS under a Multiple-sample Scenario

This section focuses on the case generated from different initial states, a common situation encountered, for example, in several biological and engineering settings, such as the case of drosophila described in Section 3.1. As detailed in the preceding subsection, it is assumed that all potential switching-output data corresponding to all possible initial states can be acquired. Thus, the switching signal sequences and their corresponding state trajectories can span the entire

switching-state pair space. Specifically, for Problem 2, the switching signal sequences $\{\sigma^{j,i}(0), \sigma^{j,i}(1), \dots, \sigma^{j,i}(r)\}$, where $j \in \{1, 2, \dots, m\}$, $i \in \{1, 2, \dots, q\}$ and $r \in \mathbb{Z}_+$, and their state trajectories $\{x^{j,i}(0), x^{j,i}(1), \dots, x^{j,i}(r)\}$ satisfy

$$\begin{aligned} \Delta_{\mu \cdot 2^n} &= \bigcup_{j=1}^m \bigcup_{i=1}^q \{(\sigma^{j,i}(0), x^{j,i}(0))\} \\ &\subseteq \bigcup_{j=1}^m \bigcup_{i=1}^q \{(\sigma^{j,i}(t), x^{j,i}(t)) \mid t \in \{0, 1, \dots, r\}\}. \end{aligned} \quad (3.16)$$

Combined with observability, the following theorem is formulated to ascertain whether SBN (3.3) is identifiable under a multiple-sample scenario.

Theorem 3.17. *SBN (3.3) is uniquely identifiable under a multiple-sample scenario if and only if it is observable.*

Proof: (Sufficiency) If SBN (3.3) is observable, then there must exist an O-switching sequence $\{\sigma^*(0), \sigma^*(1), \dots, \sigma^*(l^*)\}$. Furthermore, due to the hypothesis of enough data, we can construct switching signal sequences as follows:

$$\begin{aligned} S^{j,1}(l^*) &:= \{\sigma^{j,1}(0), \sigma^{j,1}(1), \dots, \sigma^{j,1}(l^*)\} \\ &= \{\sigma^*(0), \sigma^*(1), \dots, \sigma^*(l^*)\}, \\ S^{j,2}(l^* + 1) &:= \{\sigma^{j,2}(0), \sigma^{j,2}(1), \dots, \sigma^{j,2}(l^* + 1)\} \\ &= \{\delta_\mu^1, \sigma^*(0), \sigma^*(1), \dots, \sigma^*(l^*)\}, \\ &\vdots \\ S^{j,\mu+1}(l^* + 1) &:= \{\sigma^{j,\mu+1}(0), \sigma^{j,\mu+1}(1), \dots, \sigma^{j,\mu+1}(l^* + 1)\} \\ &= \{\delta_\mu^\mu, \sigma^*(0), \sigma^*(1), \dots, \sigma^*(l^*)\}. \end{aligned} \quad (3.17)$$

Assume that the initial states $x^{j,1}(0) = x^{j,2}(0) = x^{j,\mu+1}(0) = \delta_{2^n}^j$, where $j \in \{1, 2, \dots, 2^n\}$. Further, the corresponding output sequences are given by:

$$\begin{aligned} P^{j,1}(l^*) &= \{y^{j,1}(0), y^{j,1}(1), \dots, y^{j,1}(l^*)\}, \\ P^{j,2}(l^* + 1) &= \{y^{j,2}(0), y^{j,2}(1), \dots, y^{j,2}(l^* + 1)\} \\ &\vdots \\ P^{j,\mu+1}(l^* + 1) &= \{y^{j,\mu+1}(0), y^{j,\mu+1}(1), \dots, y^{j,\mu+1}(l^* + 1)\}. \end{aligned} \quad (3.18)$$

Let $\bar{P}^{j,i}$ denote the output sequence consisting in the last $l^* + 1$ elements of the above output sequences $P^{j,i}(l)$, where $j \in \{1, 2, \dots, 2^n\}$, $i \in \{1, 2, \dots, \mu + 1\}$, $l = l^*$

when $i = 1$ and otherwise, $l = l^* + 1$.

Since the system is observable, 2^n different output sequences $Y^1, Y^2, \dots, Y^{2^n}$ must exist in all data $\bar{P}^{j,i}$. According to the one-to-one correspondence between the output sequences and states, we consider $Y^{j'}$ as the output sequence generated from initial state $\delta_{2^n}^{j'}$, where $j' = 1, 2, \dots, 2^n$. Subsequently, by retrieving and matching all observed data $\bar{P}^{j,i}$ with $Y^{j'}$, we define

$$\begin{cases} \tilde{x}^{j,1}(0) = \delta_{2^n}^{j'}, & \text{if } \bar{P}^{j,1} = Y^{j'}, \\ \tilde{x}^{j,k}(1) = \delta_{2^n}^{j'}, & \text{if } \bar{P}^{j,k} = Y^{j'}, \\ \tilde{y}^{j,k}(0) = y^{j,k}(0), \end{cases} \quad (3.19)$$

where $j = 1, 2, \dots, 2^n$ and $k = 2, 3, \dots, \mu + 1$. For $\bar{P}^{j,i}$, when $i = 1$, $\bar{P}^{j,1} = Y^{j'}$ can determine all states of the system; when $i = k$, $\bar{P}^{j,k} = Y^{j'}$ can determine the states at $t = 1$ starting each state. This allows us to determine the evolutionary relationship between states $\delta_{2^n}^{j'}$, $j' \in \{1, 2, \dots, 2^n\}$, and their corresponding output sequences according to $\sigma^{j,2}(0)$, $\sigma^{j,3}(0)$, and $\sigma^{j,\mu+1}(0)$, that is,

$$\begin{cases} \tilde{x}^{j,k}(1) = L\sigma^{j,k}(0)\tilde{x}^{j,1}(0), \\ \tilde{y}^{j,k}(0) = H\sigma^{j,k}(0)\tilde{x}^{j,1}(0), \end{cases} \quad (3.20)$$

which identifies the structure matrices L and H , respectively.

(Necessity) Suppose that the system is not observable. In that case, there exist at least two initial states $x_0 \neq \bar{x}_0$, such that their output sequences are the same under arbitrary switching sequence. This implies that we only can find less than 2^n different output sequences from all sequences $\bar{P}^{j,i}$, then one-to-one correspondence between $Y^{j'}$ and $\delta_{2^n}^{j'}$ cannot be obtained. Consequently, at least μ columns cannot be identified, resulting in different outcomes for some columns. This contradicts the fact that the system is identifiable. Therefore, SBN (3.3) is observable. $\square$

Based on the proof of Theorem 3.17, an algorithm for determining L and H under the multiple-sample scenario can be constructed. The resulting procedure is summarized in Algorithm 3.

By analyzing the identification problem from both partial and overall perspectives, we obtain the following property.

Theorem 3.18. *SBN (3.3) may still be uniquely identifiable under a multiple-sample scenario even if some subnetworks of SBN (3.3) are unidentifiable.*

The rationality of this result is elucidated through logical reasoning and system analysis. Each subnetwork in the SBN (3.3) has its own state transition and

Algorithm 3 Identify an observable SBN (3.3) under a multiple-sample scenario

Require: $\mu + 1$ groups of switching signal data (3.17)

Ensure: Matrices L and H

- 1: Generate $\mu + 1$ groups of output data (3.18) and denote $\bar{P}^{j,i}$ as the output sequence consisting of the last $l^* + 1$ elements of sequence $P^{j,i}(l)$, where $j \in \{1, 2, \dots, 2^n\}$, $i \in \{1, 2, \dots, \mu + 1\}$; $l = l^*$ if $i = 1$, and $l = l^* + 1$ otherwise.
 - 2: **Initialization:** Set $j' \leftarrow 1$, $\mathcal{Y} \leftarrow \emptyset$.
 - 3: **for** $j = 1$ **to** 2^n **do**
 - 4: **for** $i = 1$ **to** $\mu + 1$ **do**
 - 5: **if** $\bar{P}^{j,i} \notin \mathcal{Y}$ **then**
 - 6: $Y^{j'} = \bar{P}^{j,i}$, $\mathcal{Y} = \mathcal{Y} \cup \{Y^{j'}\}$, and $j' \leftarrow j' + 1$
 - 7: **end if**
 - 8: **end for**
 - 9: **end for**
 - 10: Construct L and H according to equations (3.19) and (3.20).
-

output functions, and can be viewed as a BN. According to Theorem 3.4 in [54], a BN is identifiable if and only if it is observable. Therefore, if a subnetwork is unidentifiable, there will be at least two states within this subnetwork that cannot be distinguished based solely on output data. Switching signals allow the system to transition between different subnetworks. When the system switches from one subnetwork to another, the output mapping changes. This change can allow states that are indistinguishable in one subnetwork to become distinguishable in another. Essentially, switching signals provide additional information that can be used for system identification. It leads to a situation where the entire SBN is identifiable, even if some individual subnetworks are not.

3.5 Examples

Example 3.19. (A single-sample scenario) After several modification on Example 3.3 in [12], we obtain a new model, which is a globally switching reachable and observable SBN with 3 subnetworks, 2 state nodes, 1 output node, and

$$\begin{cases} L' = \delta_4[3, 4, 4, 2, 1, 3, 1, 2, 1, 2, 1, 1], \\ H' = \delta_2[1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 2]. \end{cases} \quad (3.21)$$

We assume that the initial state is δ_4^1 , and its O-switching sequence and R-

switching sequence are

$$\begin{cases} \{\sigma^*(0), \sigma^*(1)\} = \delta_3\{1, 3\}, \\ \{\sigma(0), \dots, \sigma(14)\} = \delta_3\{2, 1, 2, 1, 1, 2, 2, 3, 3, 1, 1, 1, 3, 1, 3\}. \end{cases}$$

According to these two sequences and construction rule (3.13), we can obtain 16 groups of switching signal sequences, which are as follows:

$$\begin{aligned} S^0(1) &= \delta_3\{1, 3\}, \\ S^1(2) &= \delta_3\{2, 1, 3\}, \\ S^2(3) &= \delta_3\{2, 1, 1, 3\}, \\ &\vdots \\ S^{15}(16) &= \delta_3\{2, 1, 2, 1, 1, 2, 2, 3, 3, 1, 1, 1, 3, 1, 3\}. \end{aligned}$$

The corresponding observed data is as follows:

$$\begin{aligned} P^0(1) &= \delta_2\{1, 1\}, \\ P^1(2) &= \delta_2\{1, 1, 1\}, \\ P^2(3) &= \delta_2\{1, 1, 2, 2\}, \\ &\vdots \\ P^{15}(16) &= \delta_2\{1, 1, 2, 1, 2, 1, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1\}. \end{aligned} \tag{3.22}$$

Based on the first step of Algorithm 2, denote the output sequence consisting of the last 2 elements of $P^i(i+1)$ by $\bar{P}^i$, where $i \in \{0, 1, \dots, 15\}$. Then 4 distinct output sequences from (3.22) can be found, that is, $Y^1 = \bar{P}^0 = \delta_2\{1, 1\}$, $Y^2 = \bar{P}^2 = \delta_2\{2, 2\}$, $Y^3 = \bar{P}^5 = \delta_2\{2, 1\}$, $Y^4 = \bar{P}^6 = \delta_2\{1, 2\}$. Regard Y^i as the output of state δ_{2n}^i . And from Steps 4 – 12 of Algorithm 2, since

$$\bar{P}^0 = Y^1, \quad \bar{P}^1 = Y^1, \quad \bar{P}^2 = Y^2, \dots, \bar{P}^{13} = Y^4, \quad \bar{P}^{14} = Y^3, \quad \bar{P}^{15} = Y^1,$$

we can obtain the state sequence

$$\{\tilde{x}(0), \dots, \tilde{x}(15)\} = \delta_4\{1, 1, 2, 1, 2, 3, 4, 2, 1, 1, 2, 3, 4, 4, 3, 1\}.$$

According to Step 13 of Algorithm 2, we know the output strategy of the state

sequence, which is

$$\{\tilde{y}(0), \tilde{y}(1), \dots, \tilde{y}(15)\} = \delta_2\{1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 2, 2, 1, 1, 2\}.$$

Finally, based on the final step of Algorithm 2, the matrices L and H can be identified

$$\begin{cases} L = \delta_4[2, 3, 4, 3, 1, 1, 4, 2, 1, 1, 1, 4], \\ H = \delta_2[1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 2, 1]. \end{cases}$$

Clearly, the identifiable matrices H and L are equivalent to those of the original system (3.21) (i.e. L' and H'), and the coordinate transformation is $\tilde{x}(t) = \delta_4[1, 3, 4, 2]x'(t)$, where $\tilde{x}(t)$ and $x'(t)$ are states of the identified system and the original system, respectively.

Example 3.20. (A multiple-sample scenario) Give an observable SBN with 2 sub-networks, 3 state nodes, 1 output node, and

$$\begin{cases} L'' = \delta_8[4, 2, 4, 3, 6, 6, 8, 7, 2, 3, 1, 1, 7, 8, 8, 5], \\ H'' = \delta_2[1, 2, 2, 1, 2, 1, 1, 2, 1, 2, 1, 2, 2, 1, 2, 1]. \end{cases} \quad (3.23)$$

Assume that the O-switching sequence of system (3.23) is

$$\{\sigma^*(0), \sigma^*(1), \dots, \sigma^*(4)\} = \delta_2\{1, 2, 1, 2, 1\}. \quad (3.24)$$

Then, we construct the switching sequences according to (3.24) and switching signal δ_2^k as

$$\begin{aligned} S^{j,1}(4) &= \delta_2\{1, 2, 1, 2, 1\}, \\ S^{j,k+1}(5) &= \delta_2\{k, 1, 2, 1, 2, 1\}, \end{aligned}$$

where $j \in \{1, 2, \dots, 8\}$ and $k = \{1, 2\}$. Let the initial state $x^{j,1}(0) = x^{j,2}(0) = x^{j,3}(0) = \delta_8^j$, and consider state δ_8^1 first. We can obtain the following output sequence groups with respect to state δ_8^1 :

$$\begin{aligned} P^{1,1}(4) &= \{1, 2, 1, 2, 1\}, \\ P^{1,2}(5) &= \{1, 1, 1, 1, 2, 1\}, \\ P^{1,3}(5) &= \{1, 2, 2, 2, 2, 1\}. \end{aligned}$$

Similarly, we can get the output sequences with respect to the remaining states as

$$\begin{aligned}
P^{2,1}(4) &= \{2, 2, 2, 2, 1\}, \\
P^{2,2}(5) &= \{2, 2, 2, 2, 2, 1\}, \\
P^{2,3}(5) &= \{2, 2, 2, 1, 2, 1\}, \\
&\vdots \\
P^{8,1}(4) &= \{2, 2, 2, 2, 2\}, \\
P^{8,2}(5) &= \{2, 1, 1, 2, 1, 2\}, \\
P^{8,3}(5) &= \{1, 2, 1, 2, 2, 2\}.
\end{aligned}$$

Let $\bar{P}^{j,i}$ be the output sequence consisting of the last 5 elements of sequence $P^{j,i}(l)$, and we have 8 distinct output sequences

$$\begin{aligned}
Y^1 &= \bar{P}^{1,1} = \{1, 2, 1, 2, 1\}, & Y^2 &= \bar{P}^{1,2} = \{1, 1, 1, 2, 1\}, \\
Y^3 &= \bar{P}^{1,3} = \{2, 2, 2, 2, 1\}, & Y^4 &= \bar{P}^{2,3} = \{2, 2, 1, 2, 1\}, \\
Y^5 &= \bar{P}^{5,1} = \{2, 1, 2, 2, 2\}, & Y^6 &= \bar{P}^{5,2} = \{1, 1, 2, 2, 2\}, \\
Y^7 &= \bar{P}^{5,3} = \{1, 1, 2, 1, 2\}, & Y^8 &= \bar{P}^{6,3} = \{2, 2, 2, 2, 2\}.
\end{aligned}$$

Denote that Y^j is the output sequence group of state $\delta_{2^n}^j$ under switching sequences (3.24). Naturally, we can find the evolutionary trajectory of each state under switching signal δ_2^1 and δ_2^2 , respectively,

$$\begin{aligned}
\delta_8^2 &= L\delta_2^1\delta_8^1, & \delta_8^3 &= L\delta_2^2\delta_8^1, & \delta_8^4 &= L\delta_2^1\delta_8^2, & \delta_8^1 &= L\delta_2^2\delta_8^2, \\
\delta_8^3 &= L\delta_2^1\delta_8^3, & \delta_8^4 &= L\delta_2^2\delta_8^3, & \delta_8^2 &= L\delta_2^1\delta_8^4, & \delta_8^1 &= L\delta_2^2\delta_8^4, \\
\delta_8^6 &= L\delta_2^1\delta_8^5, & \delta_8^7 &= L\delta_2^2\delta_8^5, & \delta_8^6 &= L\delta_2^1\delta_8^6, & \delta_8^8 &= L\delta_2^2\delta_8^6, \\
\delta_8^8 &= L\delta_2^1\delta_8^7, & \delta_8^8 &= L\delta_2^2\delta_8^7, & \delta_8^7 &= L\delta_2^1\delta_8^8, & \delta_8^5 &= L\delta_2^2\delta_8^8.
\end{aligned}$$

At the same time, we know that

$$\begin{aligned}
\delta_2^1 &= H\delta_2^1\delta_8^1, & \delta_2^1 &= H\delta_2^2\delta_8^1, & \delta_2^2 &= H\delta_2^1\delta_8^2, & \delta_2^2 &= H\delta_2^2\delta_8^2, \\
\delta_2^2 &= H\delta_2^1\delta_8^3, & \delta_2^2 &= H\delta_2^2\delta_8^3, & \delta_2^1 &= H\delta_2^1\delta_8^4, & \delta_2^1 &= H\delta_2^2\delta_8^4, \\
\delta_2^2 &= H\delta_2^1\delta_8^5, & \delta_2^2 &= H\delta_2^2\delta_8^5, & \delta_2^1 &= H\delta_2^1\delta_8^6, & \delta_2^1 &= H\delta_2^2\delta_8^6, \\
\delta_2^1 &= H\delta_2^1\delta_8^7, & \delta_2^2 &= H\delta_2^2\delta_8^7, & \delta_2^1 &= H\delta_2^1\delta_8^8, & \delta_2^1 &= H\delta_2^2\delta_8^8.
\end{aligned}$$

Thus, based on the final step of Algorithm 3, the matrices L and H can be

identified

$$\begin{cases} L = \delta_8[2, 4, 3, 2, 6, 6, 8, 7, 3, 1, 4, 1, 7, 8, 8, 5], \\ H = \delta_2[1, 1, 2, 2, 2, 1, 1, 2, 1, 2, 2, 1, 2, 1, 2, 1]. \end{cases}$$

Furthermore, the coordinate transformation is $\tilde{x}(t) = \delta_8[1, 3, 4, 2, 5, 6, 7, 8]x''(t)$.

However, based on Theorem 3.14, it is concluded that SBN (3.23) is not uniquely identifiable under a single-sample scenario due to the lack of global switching reachability. This indicates that the identification of an SBN under the multiple-sample scenario cannot imply its identification under the single-sample scenario. Additionally, the conditions required for identifying an SBN under the single-sample scenario are more stringent than those under the multiple-sample scenario.

Moreover, under a multiple-sample scenario, if considered separately, using Theorem 3.4 proposed in [54], we can know that the first subnetwork is not uniquely identifiable, whereas the second subnetwork is uniquely identifiable. Despite the presence of an unidentifiable subnetwork, the entire SBN (3.23) remains uniquely identifiable under the multiple-sample scenario.

Example 3.21. (A practical application) Consider the simple SBN model of an apoptosis network [9]:

$$\begin{cases} X_1(t+1) = f_1^{\Sigma(t)}(X_1(t), X_2(t), X_3(t)), \\ X_2(t+1) = f_2^{\Sigma(t)}(X_1(t), X_2(t), X_3(t)), \\ X_3(t+1) = f_3^{\Sigma(t)}(X_1(t), X_2(t), X_3(t)), \end{cases} \quad (3.25)$$

where $\Sigma(t) : \mathbb{N} \rightarrow \{1, 2\}$, X_1, X_2 and X_3 represent inhibitor of apoptosis proteins, active caspase 3, and active caspase 8 respectively, u represents tumor necrosis factor, and $f_1^1 = \neg X_2(t)$, $f_1^2 = 0$, $f_2^1 = f_2^2 = \neg X_1(t) \wedge X_3(t)$, $f_3^1 = 1$, $f_3^2 = X_2(t)$.

In this example, we are interested in the active caspase 3 and the state that indicates whether the cell is likely to survive or undergo apoptosis when $\Sigma(t) = 1$, and in the inhibitor of apoptosis proteins and the state that indicates whether the cell is likely to survive or undergo apoptosis when $\Sigma(t) = 2$. Then, the output equations of the subnetworks are

$$\begin{cases} Y_1(t) = h_1^{\Sigma(t)}(X_1(t), X_2(t), X_3(t)), \\ Y_2(t) = h_2^{\Sigma(t)}(X_1(t), X_2(t), X_3(t)), \end{cases} \quad (3.26)$$

where $h_1^1 = X_2(t)$, $h_1^2 = X_1(t)$, and $h_2^1 = h_2^2 = (X_1(t) \wedge X_3(t)) \vee (\neg X_1(t) \wedge X_2(t))$.

According to the construction of (3.3), we can obtain the following algebraic form

of the system (3.25) with (3.26)

$$\begin{cases} x(t+1) = L'\sigma(t)x(t), \\ y(t) = H'\sigma(t)x(t), \end{cases} \quad (3.27)$$

where $\sigma(t) \in \{1, 2\}$, and

$$\begin{cases} L' = \delta_8[7, 7, 3, 3, 5, 7, 1, 3, 7, 7, 8, 8, 5, 7, 6, 8], \\ H' = \delta_4[1, 2, 3, 4, 1, 1, 4, 4, 1, 2, 1, 2, 3, 3, 4, 4]. \end{cases}$$

Assume that the O-switching sequence of system (3.27) is

$$\{\sigma^*(0), \sigma^*(1), \sigma^*(2)\} = \delta_2\{2, 1, 1\}.$$

Similar to the identification procedure of Example 3.20, we can determine the matrices L and H , which are

$$\begin{cases} L = \delta_8[2, 1, 2, 4, 4, 4, 7, 2, 2, 8, 2, 5, 5, 5, 7, 2], \\ H = \delta_4[1, 4, 2, 3, 4, 4, 1, 1, 1, 4, 2, 1, 4, 2, 3, 3]. \end{cases}$$

Correspondingly, the coordinate transformation is $\tilde{x}(t) = \delta_8[1, 3, 4, 6, 7, 8, 2, 5]x(t)$.

3.6 Conclusion

This chapter investigated the identification problem of SBNs under both single and multiple samples. Initially, a matrix-based method was proposed to determine the observability of an SBN. This method offers advantages such as solely relying on matrix calculations and involves lower-dimensional matrix computations compared to augmented methods. Several theorems and corollaries regarding the identification of SBNs were then introduced. Two constructive procedures were established to infer the network structure, logical connections between nodes, state and output transition rules for SBNs in both single and multiple sample cases. It is noted that the output function of each subnetwork may vary with the transformation of the switching signal, rendering the measure used for investigating BCNs inapplicable to the system discussed in this chapter. Nevertheless, the proposed approach can effectively address the identification problem of traditional SBNs when the output function of each subnetwork remains uniform. Another notable finding is that even if some subnetworks of the SBN are not individually identifiable, the SBN as a whole

may still be identifiable under certain conditions due to the influence of switching signal on both states and outputs.

Chapter 4

Recurrent Output Tracking of Boolean Networks

4.1 Introduction

In the previous chapter, the identification problem for SBNs was systematically studied, where the system dynamics were reconstructed from observed data. These results lay the foundation for obtaining accurate mathematical models of logical dynamical systems. With the system models available, it is natural to proceed to the control stage. Therefore, the focus of this thesis shifts from system identification to control design. In this chapter, we investigate the ROTP for BNs and BCNs.

As is well known, the OTP is a significant problem in control theory, aiming to design control strategies such that the system outputs track a prescribed reference signal. For BCNs, various algebraic methods have been developed to address exact OT and output regulation problems [85, 86, 87, 88, 93]. However, in many practical scenarios, it is not necessary for the outputs to continuously follow the reference signal; instead, it suffices that the desired output values can be tracked intermittently over time.

In this chapter, we thus propose ROT, where the system outputs are required to track a desired reference signal intermittently and infinitely. Compared with the traditional OT, ROT relaxes the temporal constraints while still ensuring long-term performance. Leveraging the STP tool, we propose solvability criteria for ROTP in both BNs and BCNs starting from an initial state, and construct algorithms to identify all states that enable ROT of the reference signal in these systems. Additionally, a truth matrix-based method is used to design state feedback controllers, ensuring ROTP solvability for BCNs initiating from such states. Furthermore, the

recurrent periods of BNs and BCNs are discussed, demonstrating that the output tracking problem is a specific instance of the ROTP.

4.2 Problem Formulation

Recall the algebraic forms of BNs and BCNs, i.e. (2.6) and (2.8), in Chapter 2. It is noted in this chapter that for the BN (2.6), the state and output at time t starting from the initial state $x_0 \in \Delta_{2^n}$ are denoted by $x(t; x_0)$ and $y(t; x_0)$ respectively; for the BCN (2.8), the state and output at time t starting from the initial state x_0 under a control sequence $u = \{u(0), u(1), \dots, u(t-1)\}$ are denoted as $x(t; x_0, u)$ and $y(t; x_0, u)$, respectively. Additionally, we consider that the output reference signal is a constant value.

The definitions of ROT of the systems (2.6) and (2.8) starting from an initial state x_0 are given as follows.

Definition 4.1. (Recurrent Output Tracking for BNs) For a given reference signal $y^* \in \Delta_{2^p}$, the output of system (2.6) recurrently tracks y^* starting from an initial state $x_0 \in \Delta_{2^n}$, if for any $t \in \mathbb{N}$ there exists an integer $T \geq t$ such that $y(T; x_0) = y^*$.

Definition 4.2. (Recurrent Output Tracking for BCNs) For a reference signal $y^* \in \Delta_{2^p}$, the output of system (2.8) recurrently tracks y^* starting from state $x_0 \in \Delta_{2^n}$, if for any $t \in \mathbb{N}$, there exists an integer $T \geq t$ and a control sequence $u = \{u(0), u(1), \dots, u(T-1)\}$, such that $y(T; x_0, u) = y^*$.

Remark 4.3. *The above definitions characterize ROT under a given initial state, which can be regarded as a local formulation focusing on a specific initial condition. In the next chapter, this problem will be further extended to a global setting, where ROT is required to be achieved from any initial state. This progression reflects a systematic development from a simpler to a more general and stringent problem formulation.*

In the following, we first explore the conditions under which the ROTP of BNs starting from an initial state is solvable, then we study the solvability and the controller design of the ROTP for BCNs starting from an initial state. Finally, the recurrent periods of these systems are discussed.

4.3 Recurrent Output Tracking of BNs

To solve the ROTP of the BN (2.6) starting from an initial state $x_0 \in \Delta_{2^n}$, we first verify whether there exists at least one state whose output is y^* and that belongs to a cycle. The definition of a cycle of a BN is defined as follows:

Definition 4.4. ([32]) Given system (2.6), the set $\{x_0, Lx_0, \dots, L^{(d-1)}x_0\}$ is called a cycle of length d , with $d \in \{1, 2, \dots, 2^n\}$ if $L^d x_0 = x_0$ and the states in set $\{x_0, Lx_0, \dots, L^{(d-1)}x_0\}$ are pairwise different.

Cycles of length $d = 1$ are called fixed points. Evidently, a state that reaches itself at d steps will reach itself again at αd steps, where $\alpha \in \mathbb{Z}_+$. Thus, all fixed points and the states in the cycles of the system can reach themselves infinitely. By iteratively computing (2.6) starting from t until $t + d$ one obtains

$$x(t + d) = Lx(t + d - 1) = LLx(t + d - 2) = \dots = L^{(d)}x(t). \quad (4.1)$$

We define the set

$$\Psi_d := \{\delta_{2^n}^i \mid (L^{(d)})_{i,i} = 1, (L^\omega)_{i,i} \neq 1, \forall \omega < d\}. \quad (4.2)$$

Based on equation (4.1), the condition $(L^{(d)})_{i,i} = 1$ in equation (4.2) implies $(\delta_{2^n}^i)^T L^{(d)} \delta_{2^n}^i = 1$, that is $\delta_{2^n}^i = L^{(d)} \delta_{2^n}^i$. Hence for an initial state $x_0 = \delta_{2^n}^i$, $x(d; x_0) = \delta_{2^n}^i$. Condition $(L^{(\omega)})_{i,i} \neq 1, \forall \omega < d$, indicates that the state $\delta_{2^n}^i$ can not reach itself in a number of steps less than d . Thus, the set Ψ_d contains all states in cycles of length d . Further, $N_d := \frac{|\Psi_d|}{d}$ is the number of cycles of length d . Notably, if there is no cycle of length d , then $\Psi_d = \emptyset$. Otherwise, $|\Psi_d|$ must be a multiple of d . Thus, the value of N_d is a positive integer or 0.

Algorithm 4 provides a procedure to calculate all cycles of length d for system (2.6). We remark that fixed points and cycles are all called attractors.

Remark 4.5. The overall complexity of Algorithm 4 is $O(d \times N_d)$ since Step 2 needs 1 operation, the calculation of each cycle with length d $\mathcal{C}_l^d$ (Step 5) needs $d - 1$ operations and there are N_d cycles with length d , and the calculation of union (Step 6) needs $N_d - 1$ operations. Compared to the method proposed in [32] for calculating the cycles of BNs, Algorithm 4 efficiently determines cycles of length d by leveraging the power of L , rather than computing the proper factors of d .

In order to simplify the following matrix calculations, we express the attractor $\mathcal{C}_l^d$ as a column vector denoted by O_l^d , that is, $O_l^d := \sum_{\delta_{2^n}^i \in \mathcal{C}_l^d} \delta_{2^n}^i$.

Algorithm 4 Calculation of all cycles of length d of system (2.6)

Require: L, d, Ψ_d
Ensure: Cycles of length d : $\mathcal{C}_1^d, \mathcal{C}_2^d, \dots, \mathcal{C}_{N_d}^d$

```

1: Initialization: Set  $l \leftarrow 1, N_d \leftarrow 0, \mathcal{C}_0^d \leftarrow \emptyset$ .
2: Calculate the number of cycles  $N_d = \frac{|\Psi_d|}{d}$ .
3: while  $l \leq N_d$  do
4:   Choose a state  $x_0 \in \Psi_d \setminus \bigcup_{j=0}^{l-1} \mathcal{C}_j^d$ .
5:   Calculate the cycle  $\mathcal{C}_l^d \leftarrow \{x_0, Lx_0, \dots, L^{(d-1)}x_0\}$ .
6:   if  $\bigcup_{j=1}^l \mathcal{C}_j^d = \Psi_d$  then
7:     return  $\mathcal{C}_1^d, \mathcal{C}_2^d, \dots, \mathcal{C}_{N_d}^d$ 
8:     break
9:   end if
10:   $l \leftarrow l + 1$ 
11: end while
    
```

Then, assume the reference signal is $y^* = \delta_{2^p}^\beta$ and define the set $W(y^*)$ as:

$$W(y^*) = \{\delta_{2^n}^i \mid \text{Col}_i(H) = \delta_{2^p}^\beta, i \in \{1, 2, \dots, 2^n\}\}. \quad (4.3)$$

$W(y^*)$ includes all states whose output equals $\delta_{2^p}^\beta$. To make the ROTP meaningful, in the following we suppose $W(y^*) \neq \emptyset$. We express the set $W(y^*)$ in a column vector form denoted by Y^* , that is,

$$Y^* = \sum_{\delta_{2^n}^i \in W(y^*)} \delta_{2^n}^i. \quad (4.4)$$

The following theorem provides conditions for the solvability of the ROTP of system (2.6), that is, it gives the criterion to check whether the outputs of system (2.6) can recurrently track y^* starting from a given initial state.

Theorem 4.6. *The ROTP of system (2.6) starting from an initial state $x_0 = \delta_{2^n}^a$ is solvable if and only if, there exists an integer $\alpha \in \mathbb{Z}_+$ and a column vector O_l^d , where $d \in \{1, 2, \dots, 2^n\}$ and $l \in \{1, 2, \dots, N_d\}$, such that*

$$\text{Col}_a(L^\alpha) \circ O_l^d \neq \mathbf{0}_{2^n} \text{ and } O_l^d \circ Y^* \neq \mathbf{0}_{2^n}. \quad (4.5)$$

Proof: (Necessity) Since the state space of system (2.6) is finite, every state must eventually converge to an attractor. For the given initial state $x_0 = \delta_{2^n}^a$, there exists an integer $\alpha > 0$ and a cycle $\mathcal{C}_l^d$ where $d \in \{1, 2, \dots, 2^n\}$ and $l \in \{1, 2, \dots, N_d\}$ such that

$$L^\alpha \delta_{2^n}^a = \text{Col}_a(L^\alpha) \in \mathcal{C}_l^d.$$

Assume that $\mathcal{C}_l^d = \{\delta_{2^n}^{i_1}, \delta_{2^n}^{i_2}, \dots, \delta_{2^n}^{i_d}\}$. Then, there exists an integer j such that

$$\text{Col}_a(L^\alpha) = \delta_{2^n}^{i_j}.$$

This implies $\text{Col}_a(L^\alpha) \circ O_l^d = \delta_{2^n}^{i_j} \neq \mathbf{0}_{2^n}$.

Next, to prove the second condition, we proceed by contradiction. Assume that $O_l^d \circ Y^* = \mathbf{0}_{2^n}$, which implies $\mathcal{C}_l^d \cap W(y^*) = \emptyset$. In other words, the outputs of all state in the attractor $\mathcal{C}_l^d$ are not equal to y^* . Further, $y(t; \delta_{2^n}^a) \neq y^*$ for any time t . This contradicts the definition of ROT (Definition 4.1). Therefore, the assumption is false, and $O_l^d \circ Y^* \neq \mathbf{0}_{2^n}$.

(Sufficiency) The sufficiency follows directly from the necessity proof. If both conditions are satisfied, the state $x_0 = \delta_{2^n}^a$ transitions to a cycle $\mathcal{C}_l^d$ through L^α , and there exists at least one state in $\mathcal{C}_l^d$ whose output is y^* . Thus, the output of the BN starting from x_0 recurrently tracks y^* . $\square$

Next, we provide a procedure to find all states whose outputs can recurrently track y^* in BN (2.6). Such states are collected into the set represented by $\mathcal{M}$. In order to do this we firstly define suitable sequence of sets.

Assume that in BN (2.6) there are $\omega \in \mathbb{Z}_+$ attractors, denoted as $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_\omega$. For BN (2.6), given a set of states $\mathcal{C}$ such that $\mathcal{C} \neq \emptyset$, a sequence of sets associated to $\mathcal{C}$ are defined as follows:

$$\mathcal{M}_j(\mathcal{C}) = \{\delta_{2^n}^i \mid \text{Col}_i(L) \in \mathcal{M}_{j-1}(\mathcal{C}), \delta_{2^n}^i \in \Delta_{2^n} \setminus \bigcup_{l=0}^{j-1} \mathcal{M}_l(\mathcal{C})\}, \quad (4.6)$$

where $j \in \{1, 2, \dots, 2^n - 1\}$ and $\mathcal{M}_0(\mathcal{C}) = \mathcal{C}$. The set $\mathcal{M}_j(\mathcal{C})$ represents the j -th step reachable set, which includes states that can enter set $\mathcal{C}$ at the j -th step. The upper bound of j is $2^n - 1$, as the worst case occurs when system's unique attractor is a fixed point and each reachable set consists only in one state. In general, there exists an integer $\bar{j} \leq 2^n - 1$ such that $\mathcal{M}_j(\mathcal{C}) = \emptyset$ for $j > \bar{j}$.

Algorithm 5 computes the state set $\mathcal{M}$ of system (2.6). The computational complexity is $O(2^{n+1})$, where n is the number of state nodes. The complexity arises from $O(2^n)$ operations for attractors (Step 3), $O(\omega)$ for intersection checks with $W(y^*)$ (Step 4), and $O(2^n - |\mathcal{C}|)$ for iterating over reachable sets (Step 9), where ω is the number of attractors and $|\mathcal{C}|$ is the size of the current attractor set. From Step 2 to Step 7 in Algorithm 5, the algorithm checks for intersection between the attractors and $W(y^*)$. If an attractor intersect with $W(y^*)$ it is included in the set $\mathcal{C}$. Consequently, the output of (2.6) starting from any state in $\mathcal{C}$ recurrently

Algorithm 5 Calculation of set $\mathcal{M}$

Require: Matrices L and Y^* , and all attractors of system (2.6): $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_\omega$

Ensure: Set $\mathcal{M}$

```

1: Initialization: Set  $l \leftarrow 1$ ,  $j \leftarrow 1$ ,  $\mathcal{C} \leftarrow \emptyset$ , and  $\mathcal{M} \leftarrow \emptyset$ .
2: for  $l = 1$  to  $\omega$  do
3:   Calculate  $O_l \leftarrow \sum_{\delta_{2^n}^i \in \mathcal{C}_l} \delta_{2^n}^i$ .
4:   if  $O_l \circ Y^* \neq \mathbf{0}_{2^n}$  then
5:     Update the set:  $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_l$ .
6:   end if
7: end for
8: while  $j \leq 2^n - |\mathcal{C}|$  do
9:   Calculate  $\mathcal{M}_j(\mathcal{C})$  according to (4.6).
10:  if  $\mathcal{M}_j(\mathcal{C}) = \emptyset$  then
11:    Compute the set:  $\mathcal{M} \leftarrow \bigcup_{i=1}^j \mathcal{M}_i(\mathcal{C}) \cup \mathcal{C}$ .
12:    return  $\mathcal{M}$ 
13:    break
14:  end if
15:   $j \leftarrow j + 1$ 
16: end while

```

tracks y^* . From Step 8 to 15, the algorithm calculates the reachable sets of $\mathcal{C}$ and combines them with $\mathcal{C}$ to form the set $\mathcal{M}$. Specifically, the states in set $\mathcal{M}_i(\mathcal{C})$ are those that enter the set $\mathcal{C}$ at the i -th step, where $i \in \{1, 2, \dots, j\}$. Therefore, the output of (2.6) starting from any state in $\mathcal{M}_i(\mathcal{C})$ recurrently tracks y^* . Ultimately, $\mathcal{M}$ encompasses all states whose outputs can recurrently track y^* . In the special case where $\mathcal{M} = \Delta_{2^n}$, the outputs of system (2.6) recurrently track y^* from any initial states.

4.4 Recurrent Output Tracking of BCNs

In this section, our objective encompasses two aspects. The first is to determine the solvability of the ROTP for system (2.8), which involves verifying the conditions that guarantee the output of system (2.8) can recurrently track a reference signal y^* , starting from an initial state $x_0 \in \Delta_{2^n}$. The second objective, given the ROTP is solvable, is to identify all states that satisfy the conditions for implementing ROT and to design a time-invariant state feedback controller in the form of:

$$U_i(t) = \hat{g}_i(X(t)), \quad i = \{1, 2, \dots, m\}, \quad (4.7)$$

where $g_i : \mathcal{D}^n \rightarrow \mathcal{D}$, $i = \{1, 2, \dots, m\}$ is the Boolean function such that the outputs of system (2.7), starting from any of the identified states, recurrently track y^* .

According to Lemma 2.4, the state feedback controller for system (2.8) is written as

$$u(t) = \hat{G}x(t), \quad (4.8)$$

where $\hat{G} \in \mathcal{L}_{2^m \times 2^n}$ is a state feedback gain matrix.

Note that while in Definition 4.2 no assumption is made regarding the control applied to system (2.8), the following analysis focuses on state feedback control (4.8). Under this assumption, Definition 4.2 can be equivalently stated as: *For a reference signal $y^* \in \Delta_{2^p}$, the output of system (2.8) recurrently tracks y^* starting from state $x_0 \in \Delta_{2^n}$, if for any $t \in \mathbb{N}$, there exists an integer $T \geq t$ and feedback gain matrix $\hat{G}$, such that $y(T; x_0, u) = y^*$.*

Next, we present the conditions for the solvability of the ROTP of system (2.8) under the control defined in (4.8). To this end, we first introduce some definitions that are instrumental to the subsequent analysis. We split L equally into 2^m blocks as $L = [L_1, L_2, \dots, L_{2^m}]$, where $L_i \in \mathcal{L}_{2^n \times 2^n}$, and we define two matrices

$$M := \sum_{\mathcal{B}}^{2^m} L_i \in \mathcal{B}_{2^n \times 2^n} \quad (4.9)$$

and

$$F := \sum_{\mathcal{B}}^{2^n} M^{(j)} \in \mathcal{B}_{2^n \times 2^n}. \quad (4.10)$$

The matrix M includes all possible one-step state transition information, that is, if state $\delta_{2^n}^i$ is reachable from state $\delta_{2^n}^j$ in one step, then $(M)_{i,j} = 1$; otherwise, $(M)_{i,j} = 0$. Similarly, the matrix F encodes all possible state transition information across any number of steps, that is, if state $\delta_{2^n}^i$ is reachable from state $\delta_{2^n}^j$, then $(F)_{i,j} = 1$; otherwise, $(F)_{i,j} = 0$.

Given the reference signal $y^* = \delta_{2^p}^\beta$, we define a set

$$\Theta := \{\delta_{2^n}^i \mid (F)_{i,i} = 1, \delta_{2^n}^i \in W(y^*)\}, \quad (4.11)$$

where $W(y^*)$ has been defined in (4.3). Obviously, all states in set Θ are reachable from themselves and their outputs are equal to y^* , that is, they are recurrent states whose output is y^* . Unlike the analysis of BNs, finding all attractors of a BCN, especially cycles, is a demanding task. Thus, in order to solve the ROTP for system (2.8) easily, in what follows we will only focus on set Θ rather than calculating the

cycles that contains the states in Θ . Firstly, we define the vector form of set Θ as

$$V^* := \sum_{\delta_{2^n}^i \in \Theta} \delta_{2^n}^i.$$

Next, in the following theorem we give the conditions for solvability of the ROTP of (2.8) starting from an initial state x_0 .

Theorem 4.7. *The ROTP of system (2.8) starting from an initial state $x_0 = \delta_{2^n}^a$ is solvable if and only if the following conditions hold:*

- (i) $V^* \neq \mathbf{0}_{2^n}$;
- (ii) $\text{Col}_a(F) \circ V^* \neq \mathbf{0}_{2^n}$.

Proof: (Sufficiency) Since $V^* \neq \mathbf{0}_{2^n}$, we know that $\Theta \neq \emptyset$. It implies that there exists at least one state $\delta_{2^n}^i$ in Θ , and according to (4.11), $(F)_{i,i} = 1$ and $\delta_{2^n}^i \in W(y^*)$ both hold. Then for the state $\delta_{2^n}^i$, there exists an integer $\rho' \in \mathbb{Z}_+$ and a control sequence $u^0 = \{u(0), u(1), \dots, u(\rho' - 1)\}$ such that

$$\begin{cases} x(\rho'; \delta_{2^n}^i, u^0) = \delta_{2^n}^i, \\ H\delta_{2^n}^i = y^*. \end{cases}$$

Therefore, for any $t \in \mathbb{Z}_+$, there exists an integer $\tilde{\rho} = \alpha \times \rho' \geq t$, where $\alpha \in \mathbb{Z}_+$, and a control sequence $\tilde{u}^0 = \{u(0), u(1), \dots, u(\rho' - 1), u(\rho'), \dots, u(\tilde{\rho} - 1)\}$ such that

$$y(\tilde{\rho}; \delta_{2^n}^i, \tilde{u}^0) = y^*,$$

which implies that the outputs of system (2.8) starting from state $\delta_{2^n}^i$ can recurrently track y^* . Obviously, $\{u((\alpha - 1)\rho'), u((\alpha - 1)\rho' + 1), \dots, u(\alpha\rho' - 1)\} = \tilde{u}^0$ for any $\alpha \in \{1, 2, \dots, \frac{\tilde{\rho}}{\rho'}\}$. Without loss of generality, for any state $\delta_{2^n}^i \in \Theta$, the outputs of system (2.8) starting from the states can recurrently track y^* .

Then for condition (ii), without loss of generality we assume that $\text{Col}_a(F) \circ V^* = \delta_{2^n}^b$, which implies

$$\begin{cases} \text{Col}_a(F) = \delta_{2^n}^b, \\ \delta_{2^n}^b \in \Theta. \end{cases} \quad (4.12)$$

$\text{Col}_a(F) = \delta_{2^n}^b$ indicates that there exists at least one finite integer $\bar{\rho} \leq 2^n$ and a control sequence $\bar{u} = \{u(0), u(1), \dots, u(\bar{\rho} - 1)\}$, such that $x(\bar{\rho}, \delta_{2^n}^a, \bar{u}) = \delta_{2^n}^b$. Since $\delta_{2^n}^b \in \Theta$, for state $\delta_{2^n}^a$ and any $t \in \mathbb{Z}_+$, there exists $\rho = \bar{\rho} + \tilde{\rho} \geq t$ and $u = \{u(0), u(1), \dots, u(\bar{\rho} - 1), u(\bar{\rho}), \dots, u(\rho)\}$, where $\{u(\bar{\rho}), \dots, u(\rho)\} = \tilde{u}^0$, such

that

$$y(\rho; \delta_{2^n}^i, u) = y^*,$$

which implies that the outputs of (2.8) starting from state $\delta_{2^n}^a$ can recurrently track y^* .

(Necessity) We prove it by contradiction. Assume condition (i) or (ii) does not hold.

If $V^* = \mathbf{0}_{2^n}$, i.e., $\Theta = \emptyset$, there will be two cases. The first one is that $(F)_{i,i} = 0$ for any state $\delta_{2^n}^i \in W(y^*)$, that is, for any $t \in \mathbb{Z}_+$ and any $u = \{u(0), u(1), \dots, u(t-1)\}$, we have $x(t; \delta_{2^n}^i, u) \neq \delta_{2^n}^i$, where $\delta_{2^n}^i \in W(y^*)$. This implies that it does not exist a recurrent state whose output equals y^* . Then, $y(t; \delta_{2^n}^i, u) \neq y^*$ for any integer $t > 2^{m+n}$. However, it is worth pointing out that there might be a time instant $t \in \{1, 2, \dots, 2^{m+n}\}$ such that $y(t; \delta_{2^n}^i, u) = y^*$, as the state $\delta_{2^n}^i$ can reach another state in set $W(y^*)$.

Another case is that for any state $\delta_{2^n}^i$ satisfying $(F)_{i,i} = 1$, we have $\delta_{2^n}^i \notin W(y^*)$, which implies that for system (2.8), all recurrent states outputs are not equal to y^* . In other words, for any such $\delta_{2^n}^i$, any $t > 2^{m+n}$, and any control sequence $u = \{u(0), \dots, u(t-1)\}$, we always have $y(t; \delta_{2^n}^i, u) \neq y^*$.

If $\text{Col}_a(F) \circ V^* = \mathbf{0}_{2^n}$, according to the analysis of sufficiency, we know that there is no integer $b \in \{1, 2, \dots, 2^n\}$ satisfying (4.12), which means that for the state $\delta_{2^n}^a$, any $t > 2^{m+n}$, and any control sequence $u = \{u(0), \dots, u(t-1)\}$, we have $y(t; \delta_{2^n}^a, u) \neq y^*$.

Both two hypotheses contradict Definition 4.2, hence the conclusion holds. $\square$

In what follows, we propose a method to find all states whose outputs can recurrently track y^* , and we collect such states in a set denoted with $\overline{\mathcal{M}}$. To this end, we firstly consider reachable sets of Θ via the concept of truth matrices [94].

Denote $E_0 = \Theta$ and $\Omega_0 = \Delta_{2^n}$. A truth matrix $T_{E_0|\Omega_0} \in \mathcal{B}_{2^m \times 2^n}$ is given by

$$(T_{E_0|\Omega_0})_{i,j} = \begin{cases} 1, & \text{if } L_i \delta_{2^n}^j \in E_0, \forall \delta_{2^n}^j \in \Omega_0, \\ 0, & \text{otherwise.} \end{cases} \quad (4.13)$$

Let $E_1 := \{\delta_{2^n}^j \mid \text{Col}_j(T_{E_0|\Omega_0}) \neq \mathbf{0}_{2^m}\}$. For any $\delta_{2^n}^j \in E_1$, there exists at least a control such that E_0 is reachable from $\delta_{2^n}^j$ in one step. That is, E_1 is the one-step reachable set of Θ .

Similarly, we construct truth matrix $T_{E_t|\Omega_t} \in \mathcal{B}_{2^m \times 2^n}$ as

$$(T_{E_t|\Omega_t})_{i,j} = \begin{cases} 1, & \text{if } L_i \delta_{2^n}^j \in E_t, \forall \delta_{2^n}^j \in \Omega_t, \\ 0, & \text{otherwise,} \end{cases} \quad (4.14)$$

where $\Omega_t = \Omega_{t-1} \setminus E_t$. Denote $E_{t+1} := \{\delta_{2^n}^j \mid \text{Col}_j(T_{E_t|\Omega_t}) \neq \mathbf{0}_{2^m}\}$, then for any $\delta_{2^n}^j \in E_{t+1}$, there exists at least a control such that E_t is one step reachable from $\delta_{2^n}^j$. By recursive calculations, we know that state $\delta_{2^n}^j$ can reach E_0 by $t+1$ steps under proper control sequences. E_{t+1} represents the $(t+1)$ -th step reachable set.

We give the following procedure to find set $\overline{\mathcal{M}}$.

Algorithm 6 Calculation of set $\overline{\mathcal{M}}$

Require: Matrix L

Ensure: Set $\overline{\mathcal{M}}$

- 1: Calculate Θ according to (4.9), (4.10) and (4.11).
 - 2: **Initialization:** Set $t \leftarrow 1$, $E_0 \leftarrow \Theta$, and $\Omega_0 \leftarrow \Delta_{2^n}$.
 - 3: **while** $t \leq 2^n$ **do**
 - 4: Construct the truth matrix $T_{E_{t-1}|\Omega_{t-1}}$.
 - 5: Calculate $E_t \leftarrow \{\delta_{2^n}^j \mid \text{Col}_j(T_{E_{t-1}|\Omega_{t-1}}) \neq \mathbf{0}_{2^m}\}$.
 - 6: **if** $E_t = \emptyset$ **then**
 - 7: Set $t^* \leftarrow t$, and compute $\overline{\mathcal{M}} \leftarrow \bigcup_{i=0}^{t^*} E_i$.
 - 8: **return** $\overline{\mathcal{M}}$
 - 9: **break**
 - 10: **end if**
 - 11: Compute $\Omega_t \leftarrow \Omega_{t-1} \setminus E_t$.
 - 12: $t \leftarrow t + 1$
 - 13: **end while**
-

Since $(F)_{i,i} = 1$, the state $\delta_{2^n}^i$ can reach itself under proper controls, then it is a recurrent state. Obviously, the elements in set Θ are recurrent states whose outputs equal $\delta_{2^p}^\beta$, which implies that the ROTP of system (2.8) is equivalent to checking whether all states in set $\overline{\mathcal{M}}$ can reach set Θ . Similarly to Algorithm 5, the computational complexity of Algorithm 6 is $O(2^{n+1})$ due to the recursive construction of reachable sets and truth matrices. Each iteration of reachable set construction involves operations proportional to the size of the state space. In the special case where $\overline{\mathcal{M}} = \Delta_{2^n}$ the ROTP for system (2.8) is solvable from any initial state.

Remark 4.8. It is evident that $E_1 \neq \emptyset$ since the states in set Θ satisfy $(F)_{i,i} = 1$. Furthermore, if $\Theta \subseteq E_1$ and $\overline{\mathcal{M}} = \Delta_{2^n}$ then the ROTP of system (2.8) reduces to

the traditional OTP.

Next, we propose an effective method to design the state feedback control $u(t) = \hat{G}x(t)$. According to the construction of matrices $T_{E_t|\Omega_t}$ from (4.13) and (4.14) and building up on Algorithm 6, the calculation of the matrix $\hat{G}$ is proposed in the following algorithm.

Algorithm 7 Calculation of the matrix $\hat{G}$

Require: Matrix L

Ensure: Matrix $\hat{G}$

- 1: Run Algorithm 6 and get output $T_{E_{t-1}|\Omega_{t-1}}$, where $t \in \{1, 2, \dots, t^*\}$.
- 2: Compute the accumulated truth matrix $\bar{T} = \sum_{t=1}^{t^*} T_{E_{t-1}|\Omega_{t-1}}$.
- 3: Construct the state feedback gain matrix $\hat{G}$ such that

$$\begin{cases} \hat{G}|_{\bar{\mathcal{M}}} \leq \bar{T}|_{\bar{\mathcal{M}}}, \\ \hat{G}|_{\Delta_{2^n} \setminus \bar{\mathcal{M}}} \leq \mathbf{1}_{2^m \times 2^n}|_{\Delta_{2^n} \setminus \bar{\mathcal{M}}}, \end{cases} \quad (4.15)$$

4: **return** $\hat{G}$

Algorithm 7 directly provides the state feedback controllers that ensure the outputs of system (2.8), starting from the states in set $\bar{\mathcal{M}}$, recurrently track the reference signal. From the construction process of matrix $\hat{G}$, it is clear that under these state feedback controllers the system tracks the signal y^* at the first time in the shortest time. Indeed, for any state $\delta_{2^n}^j \in \bar{\mathcal{M}}$, if $\text{Col}_j(\bar{T}) \neq \mathbf{0}_{2^m}$, then there exists a positive integer l such that $\delta_{2^n}^j \in E_l$. Moreover, if $(\bar{T})_{i,j} = 1$, the control that steers $\delta_{2^n}^j$ to the set E_{l-1} can be selected as $\delta_{2^m}^i$. The output of all states $\delta_{2^n}^j \in \Delta_{2^n} \setminus \bar{\mathcal{M}}$ can not recurrently track y^* under any control sequence. Thus, we assign arbitrary values from Δ_{2^m} to the corresponding columns of $\hat{G}$ as indicated by the second formula in (4.15). Using this approach, we can design all state feedback controllers to ensure the ROT of system (2.8) starting from the states in $\bar{\mathcal{M}}$. The computational complexity is $O(t^*)$ since the controller design only requires calculating the truth matrix $\bar{T}$, which involves $O(t^*)$ operations.

It is worth noting that in [86] authors transform the OTP of BCNs into an optimal control problem, employing a cost function to derive an optimal control law based on static state feedback. Compared to optimization-based methods suggested in [86], our approach eliminates the need for computing a cost function, thus providing a more computationally efficient solution. Although our study does not address the optimal control framework for ROTP, this constitutes a promising direction for future research. In fact, the proposed method for solving the ROTP provides a

comprehensive framework to identify all states satisfying ROT and the state feedback controller that can recurrently track the desired fixed signal, thereby laying the groundwork for potential extensions to optimal control design. Additionally, to solve the OTP for BCNs, in [87] authors construct reachable sets by first retrieving the entire state space every time, and design state feedback controllers by iteratively analyzing each state in the reachable set and determining the appropriate control based on state evolution. In contrast, the proposed method constructs reachable sets while integrating them with truth matrix methods. This integration enables the direct derivation of state feedback controllers from the truth matrix, avoiding iterative state-based computations. Consequently, our approach simplifies the process and improves computational efficiency.

4.5 Recurrent Periods of BNs and BCNs

In this section, we discuss the recurrent periods of systems (2.6) and (2.8).

Definition 4.9. For a state $x_0 = \delta_{2^n}^i \in \Delta_{2^n}$ of system (2.6), $P_i \in \mathbb{Z}_+$ is called the recurrent period of state $\delta_{2^n}^i$, if an integer $T_i \geq 0$ can be found such that

$$y(T_i + \gamma P_i; x_0) = y^* \quad (4.16)$$

holds for any integer $\gamma \geq 0$. The smallest integer P_i^* making (4.16) hold is called the minimal positive recurrent period of state x_0 .

When the ROTP of system (2.6) starting from initial state $\delta_{2^n}^a$ is solvable, we can calculate its recurrent period by two steps. The first step is to calculate all attractors of system (2.6) via Algorithm 4 and find the attractor $\mathcal{C}_l^d$ that satisfies (4.5). The second step is to compute the corresponding output trajectory of the attractor according to the second formula of (2.6). Once the output trajectory is obtained, the recurrent period with respect to state $\delta_{2^n}^a$ can be obtained directly.

If the ROTP of system (2.6) is solvable, $P := \max_{1 \leq i \leq 2^n} P_i^*$ is called the recurrent period of system (2.6). Based on the process of calculating recurrent period of one state, we can obtain P after obtaining the output trajectories of all attractors.

Remark 4.10. Note that for system (2.6) the recurrent period may not be equal to the maximum length of the cycles. The reason for this is related to the output trajectory of the system. We will illustrate this concept through the following toy example; consider a BN defined by matrices $L = \delta_8[2, 3, 4, 5, 6, 1, 3, 5]$, and

$H_1 = \delta_4[1, 2, 1, 3, 4, 2, 2, 3]$ and another one having the same L but with $H_2 = \delta_4[1, 2, 1, 2, 1, 2, 2, 3]$. Assume that the reference signal is $y^* = \delta_4^1$. Since both systems have the same L they have the same longest cycle which is: $\delta_8^1 \rightarrow \delta_8^2 \rightarrow \delta_8^3 \rightarrow \delta_8^4 \rightarrow \delta_8^5 \rightarrow \delta_8^6 \rightarrow \delta_8^1$. However, the output trajectory for system L and H_1 is $\delta_4^1 \rightarrow \delta_4^2 \rightarrow \delta_4^1 \rightarrow \delta_4^3 \rightarrow \delta_4^4 \rightarrow \delta_4^2 \rightarrow \delta_4^1$ (depicted as Case 1 in Figure 4.1), with a recurrent period $P = 6$, equal to the length of the cycle. Differently, for system L and H_2 the cycle output trajectory is $\delta_4^1 \rightarrow \delta_4^2 \rightarrow \delta_4^1 \rightarrow \delta_4^2 \rightarrow \delta_4^1 \rightarrow \delta_4^2 \rightarrow \delta_4^1$ (depicted as Case 2 in Figure 4.1), and the recurrent period is $P = 2$ which is not the length of the cycle.

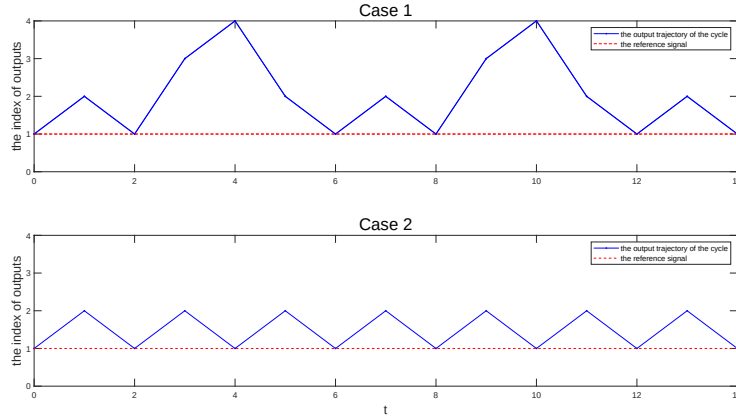


Figure 4.1: The output trajectory of the cycle: Case 1 and Case 2

Next, we consider the case of adding controls in BNs, and further give the corresponding definition of the recurrent period.

Definition 4.11. For a state $x_0 = \delta_{2^n}^i \in \Delta_{2^n}$ of system (2.8), P_i is called the recurrent period of the state $\delta_{2^n}^i$, if an integer $T_i \geq 0$ and a control sequence u^i can be found such that

$$y(T_i + aP_i; x_0, u^i) = y^* \quad (4.17)$$

holds for any integer $a \geq 0$. The smallest integer P_i^* making (4.17) hold is called the minimal positive recurrent period of state x_0 .

Since

$$x(t+1) = Lu(t)x(t) = LGx(t)x(t) = LGM_{r,2^n}x(t) := \tilde{L}x(t),$$

a BCN under a given state feedback control can be viewed as a BN. When the ROTP of system (2.8) from initial state x_0 is solvable under the controller $u(t) = Gx(t)$,

we can calculate the recurrent period of x_0 using the same method as system (2.6). And if the ROTP of system (2.8) is solvable, $P := \max_{1 \leq i \leq 2^n} P_i^*$ is called the recurrent period of system (2.6).

Remark 4.12. *It is worth noticing that the ROTP becomes the OTP when the recurrent period is $P = 1$. Therefore, the OTP discussed in reference [87] is a special case of the ROTP. In addition, periodic output regulation problem involves designing controllers such that the system tracks a periodic reference signal. In our context, when the recurrent period of BCNs is fixed and greater than 1, and there is only one cycle reachable from the system's states, the output trajectory that BCNs track under the controller given by Algorithm 7 will be periodic. Therefore, in this case the ROTP proposed in this paper can be viewed as the periodic output regulation problem.*

4.6 Examples

Example 4.13. (ROTP of BNs) Consider the BN proposed in [32] and reported here:

$$\begin{cases} x_1(t+1) &= X_3(t) \wedge X_2(t), \\ X_2(t+1) &= \neg X_1(t), \\ X_3(t+1) &= X_3(t) \vee X_2(t), \\ Y(t) &= \neg X_2(t) \wedge X_3(t), \end{cases} \quad (4.18)$$

where $X_1, X_2, X_3 \in \mathcal{D}$ are the states and $Y \in \mathcal{D}$ is the output. Assume that system (4.18) evolves starting from initial state $X(0) = (0, 1, 0)$ and the reference signal is $Y = 1$.

Based on the matrix expression, system (4.18) is written as

$$\begin{cases} x(t+1) = Lx(t), \\ y(t) = Hx(t), \end{cases} \quad (4.19)$$

where $x(t) = \times_{i=1}^3 x_i(t) \in \Delta_8$, $y \in \Delta$, $L = \delta_8[3, 7, 7, 8, 1, 5, 5, 6]$, and $H = \delta_2[2, 2, 1, 2, 2, 2, 1, 2]$. The initial state is $x(0) = \delta_8^6$ and the reference signal is $y^* = \delta_2^1$. Figure 4.2 shows the state transition graph of system (4.19).

Based on (4.3), $W(\delta_2^1) = \{\delta_8^i \mid \text{Col}_i(H) = \delta_2^1, i \in \{1, 2, \dots, 8\}\} = \{\delta_8^3, \delta_8^7\}$ and it is highlighted with a dotted circle in Figure 4.2. This implies $Y^* = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T$. Applying (4.2), we obtain $\Psi_d = \emptyset$ for $d = 1, 2, 3$ and $\Psi_4 = \{\delta_8^1, \delta_8^3, \delta_8^5, \delta_8^7\}$. Clearly,

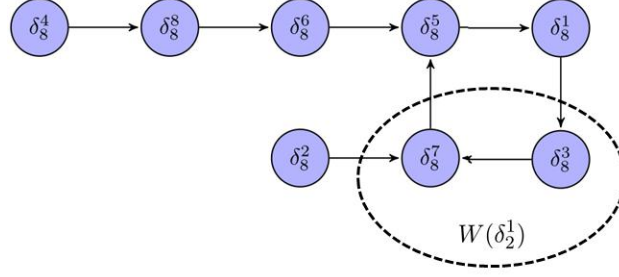


Figure 4.2: The state transition diagram of system (4.19). The blue circle represents the state and the dotted circle represents the set of states whose output equals y^* .

$|\Psi_4| = 4$, then $N_4 = \frac{|\Psi_4|}{4} = 1$, thus there exists one cycle of length 4 for system (4.19). Since the system has 8 states and only one cycle of length 4, no cycles of length greater than 4 exist.

According to Algorithm 4, we select a state $\delta_8^1 \in \Psi_4$ and compute $L\delta_8^1 = \delta_8^3$, $L^2\delta_8^1 = \delta_8^7$, $L^3\delta_8^1 = \delta_8^5$ and $L^4\delta_8^1 = \delta_8^1$. This results in $\mathcal{C}_1^4 = \{\delta_8^1, \delta_8^3, \delta_8^7, \delta_8^5\}$. Thus, we identify the unique cycle $\delta_8^1 \rightarrow \delta_8^3 \rightarrow \delta_8^7 \rightarrow \delta_8^5 \rightarrow \delta_8^1$. Consequently, $O_1^4 = [1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0]^T$. Since $\text{Col}_6(L) \circ O_1^4 \neq \mathbf{0}_{2n}$ and $O_1^4 \circ Y^* = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T \neq \mathbf{0}_{2n}$, according to Theorem 4.6, the outputs of system (4.19) can recurrently track δ_2^1 from the state δ_8^6 .

Based on Algorithm 5, since $C_1^4 \circ Y^* \neq \mathbf{0}_{2n}$, we have $\mathcal{C} = \mathcal{C}_1^4$. Next, we compute $\mathcal{M}_1(\mathcal{C}) = \{\delta_8^2, \delta_8^6\}$, $\mathcal{M}_2(\mathcal{C}) = \{\delta_8^8\}$, $\mathcal{M}_3(\mathcal{C}) = \{\delta_8^4\}$. Therefore, $\mathcal{M} = \mathcal{C} \cup \mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3 = \Delta_{2n}$. Since $\mathcal{M} = \Delta_{2n}$, we conclude that the outputs of system (4.19) can recurrently track δ_2^1 .

Note that all state trajectories of system (4.19) eventually reach and remain in cycle $\mathcal{C}$. If the output of each state in this cycle is equal to y^* , that is, the outputs do not change after tracking the reference signal, the system achieves the traditional OT. However, in this example the output trajectory of the cycle $\mathcal{C}$ is $\delta_2^2 \rightarrow \delta_2^1 \rightarrow \delta_2^1 \rightarrow \delta_2^2$ where the reference signal y^* appears recurrently. Consequently, the system does not achieve traditional OT; in fact it recurrently tracks y^* .

The efficacy of the method proposed in Section 4.4 is shown in the following biological example .

Example 4.14. (ROTP of BCNs) Consider a deduced model for the lac operon

Denote $E_0 = \Theta$, $\Omega_0 = \Delta_{2^n}$. According to Algorithm 6, we can establish the truth matrix

$$T_{E_0|\Omega_0} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathcal{B}_{8 \times 8}.$$

E_1 can be easily calculated, that is, $E_1 = \{\delta_8^4, \delta_8^8\}$.

Due to $E_1 \neq \emptyset$, then calculate $\Omega_1 = \Omega_0 \setminus E_1 = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^5, \delta_8^6, \delta_8^7\}$, and construct the truth matrix

$$T_{E_1|\Omega_1} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \in \mathcal{B}_{8 \times 8}.$$

According to the above matrix, it shows that $E_2 = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^5, \delta_8^6, \delta_8^7\}$. Then $\Omega_2 = \Omega_1 \setminus E_2 = \emptyset$, which implies $E_3 = \emptyset$.

Obviously, $\bar{\mathcal{S}} = E_1 \cup E_2 = \Delta_8$. That is, the ROTP of system (4.20) is solvable. Then calculate

$$\bar{T} = T_{E_0|\Omega_0} + T_{E_1|\Omega_1} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

according to Algorithm 7, further, the state feedback controller $\hat{G}$ to steer all states in set $\bar{\mathcal{S}}$ to Θ can be designed as

$$\hat{G} \mid_{\bar{\mathcal{S}}} \leq \bar{T} \mid_{\bar{\mathcal{S}}}.$$

We know that all possible state feedback matrices have the following forms:

$$\hat{G} = \delta_8[j_1, j_2, j_3, j_4, j_5, j_6, j_7, j_8],$$

where $j_1, j_2, j_3 \in \{1, 2, 3, 4\}$, $j_4 \in \{5, 6, 7, 8\}$, $j_5, j_6, j_7 \in \{1, 2, 3, 4, 7, 8\}$, $j_8 \in \{5, 6\}$. Consequently, the outputs of system (4.21) can recurrently track the reference signal δ_2^1 under 110592 state feedback controls. For example, we can take $\hat{G} = \delta_8[1, 1, 1, 5, 1, 1, 1, 5]$, which corresponds to the following controller:

$$\begin{cases} B_1(t) = A_2 \vee A_3, \\ B_2(t) = 1, \\ B_3(t) = 1. \end{cases} \quad (4.22)$$

Example 4.15. (Recurrent period) Recall Example 4.14.

The ROTP of system (4.21) is solvable under the state feedback controller (4.22). Then its state transition diagram is displayed in Figure 4.3.

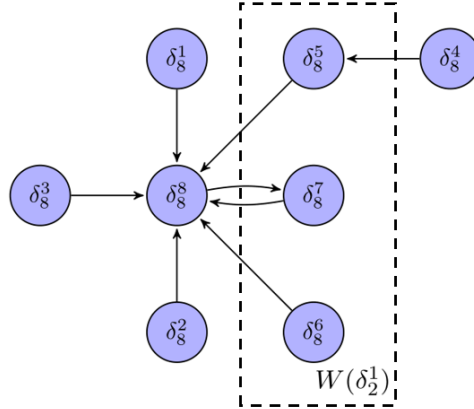


Figure 4.3: The state transition diagram of system (4.21) with the state feedback controller (4.22). The blue circle represents the state and the dotted rectangle represents the set of states whose output equals y^* .

Figure 4.3 displays that there is only one cycle $\delta_8^7 \rightarrow \delta_8^8 \rightarrow \delta_8^7$, and the corresponding output trajectory is $\delta_2^1 \rightarrow \delta_2^2 \rightarrow \delta_2^1$. It follows that the recurrent period of system (4.21) under state feedback controller (4.22) is $P = 2$.

4.7 Conclusion

This chapter first proposed and investigated the ROTP of BNs and BCNs. For BNs, by analyzing the attractor structures, necessary and sufficient conditions were established to characterize the solvability of the ROTP. For BCNs, algebraic criteria and corresponding algorithms were developed based on the truth matrix representation to determine whether the desired tracking can be achieved. Furthermore, state feedback controllers were constructed to guarantee ROT, and the notion of recurrent periods was introduced to characterize the temporal properties of the resulting trajectories. It was also shown that the traditional OTP can be viewed as a special case of the ROTP.

However, the computational complexity of the algorithms limits their applicability to smaller systems, as the dimension of the truth matrix grows exponentially with the network size. In addition, this chapter focuses on constant reference signals, while the extension to time-varying reference signals remains an open problem for future research.

Chapter 5

Recurrent Output Tracking of Switched Boolean Control Networks

5.1 Introduction

The ROTP for BNs and BCNs has been investigated in the previous chapter, where algebraic conditions and controller design methods were developed to ensure that the system outputs track a desired reference signal intermittently and infinitely. These results provide a flexible framework for relaxing conventional OT requirements and capturing long-term behaviors in logical dynamical systems.

Compared with B(C)Ns, the presence of switching significantly increases the complexity of system dynamics, since the system evolution depends not only on the control inputs but also on the switching sequence. Consequently, the controller design methods developed for B(C)Ns cannot be directly extended to SBCNs, as the ROTP requires the joint design of control inputs and switching strategies. In addition, the previous chapter focused on systems with a fixed initial state, resulting in a local formulation of the ROTP. Extending the problem to arbitrary initial states naturally leads to a global formulation, which introduces additional theoretical challenges.

Motivated by this, this chapter studies the ROTP for SBCNs. By introducing the concepts of recurrent states, the problem can be viewed as a set controllability problem, further necessary and sufficient conditions are established for the solvability of the ROTP for SBCNs. In addition, an efficient algorithm is developed to design state feedback controllers and state feedback switching signals such that the system

can recurrently track a desired reference output.

5.2 Problem Formulation

Recall the SBCN (2.12) in Chapter 2. The constant reference output signal considered in the chapter is denoted by $\tilde{Y} = (Y_1, Y_2, \dots, Y_p) \in \mathcal{D}^p$. We now extend the ROT definitions in the previous chapter to ROT for system (2.12) under a global setting.

Definition 5.1. Given system (2.12) and a desirable output signal $\tilde{Y}$, the outputs of system (2.12) are said to recurrently track $\tilde{Y}$, if for any initial state $X(0) \in \mathcal{D}^n$ and any integer $t \in \mathbb{N}$, there exist an integer $T \geq t$, a switching signal sequence $\Sigma = \{\Sigma(0), \Sigma(1), \dots, \Sigma(T-1)\}$ and a control sequence $U = \{U(0), U(1), \dots, U(T-1)\}$, such that

$$Y(T; X(0), \Sigma, U) = \tilde{Y}. \quad (5.1)$$

Remark 5.2. In Chapter 4, we proposed the definition of ROT for BNs and BCNs under a given initial state. In contrast, this chapter considers SBCNs and addresses the problem in a global sense, requiring ROT to be achieved from any initial state. This results in a more stringent formulation, as the control and switching strategies must guarantee the desired tracking performance uniformly over the entire state space.

Remark 5.3. In terms of each initial state, if for any time $t \in \mathbb{N}$, we can always find a time $T \geq t$ such that the outputs of system (2.12) at time T reach the signal $\tilde{Y}$, it is obvious that the outputs can track $\tilde{Y}$ often in a long evolutionary process. Indeed, for system (2.12), the definition still holds if the condition “for any $t \geq 0$ ” is omitted. Suppose that, for a fixed initial state X , there exist an integer $T_0 \geq 0$, a switching signal sequence $\Sigma^0 = \{\Sigma^0(0), \Sigma^0(1), \dots, \Sigma^0(T_0-1)\}$ and a control sequence $U^0 = \{U^0(0), U^0(1), \dots, U^0(T_0-1)\}$, such that state $X(0)$ evolves to $\bar{X} := X(T_0)$ and $Y(T_0; X, \Sigma^0, U^0) = \tilde{Y}$. For the state $\bar{X}$ (considered as an initial state), there exist an integer $T_1 \geq 0$, a sequence $\Sigma^1 = \{\Sigma^1(0), \Sigma^1(1), \dots, \Sigma^1(T_1-1)\}$ and a control sequence $U^1 = \{U^1(0), U^1(1), \dots, U^1(T_1-1)\}$, such that $Y(T_1; \bar{X}, \Sigma^1, U^1) = \tilde{Y}$. Then, for the state X , there exist an integer $T' := T_0 + T_1$, a sequence $\Sigma' = \{\Sigma^0, \Sigma^1\}$ and a sequence $U' = \{U^0, U^1\}$, such that $Y(T'; X, \Sigma', U') = \tilde{Y}$.

Thus, due to the existence of switching and control sequences satisfying (5.1) for each initial state, during the continuous evolution process, the outputs of system (2.12), starting from the initial state X , can recurrently equal $\tilde{Y}$. Due to the arbitrariness of the initial state, the outputs of system (2.12) recurrently track $\tilde{Y}$.

The definition we propose in this chapter is the most general and applies to any case, including systems with time-varying restricted control sets and switching signal sets. We provide this unified form to facilitate convenient and consistent study in future research.

In this chapter, we have the following two objectives:

- i) To check whether there exist a switching signal sequence and a control sequence such that the outputs of system (2.12) can track a constant output signal $\tilde{Y}$ recurrently (The solvability of the ROTP).
- ii) To design a state feedback switching signal in the form of

$$\Sigma(t) = w(X(t)), \quad (5.2)$$

and a state feedback controller in the form of

$$U_k(t) = \tilde{g}_k(X(t)), \quad k = 1, 2, \dots, m, \quad (5.3)$$

where $w : \mathcal{D}^n \rightarrow \{1, 2, \dots, \mu\}$ and $\tilde{g}_k : \mathcal{D}^n \rightarrow \mathcal{D}$, $k = 1, 2, \dots, m$ are the Boolean functions, such that the outputs of system (2.12) recurrently track $\tilde{Y}$.

The algebraic form of the SBCN (2.12) is given in Chapter 2, that is, the equation (2.13). In terms of system (2.13), the equivalent form of Definition 5.1 is given as follows.

Definition 5.4. Given a desirable output signal $\tilde{y}$, the outputs of system (2.13) are said to recurrently track $\tilde{y}$, if for any initial state x_0 and any integer $t \in \mathbb{N}$, there exist a $T \in \mathbb{N}$ with $T \geq t$, a switching signal sequence $\sigma = \{\sigma(0), \sigma(1), \dots, \sigma(T-1)\}$ and a control sequence $u = \{u(0), u(1), \dots, u(T-1)\}$, such that $y(T; x_0, \sigma, u) = \tilde{y}$.

Our objectives can be restated referring to system (2.13) as follows: i) Check whether there exist σ and u such that the outputs of system (2.13) can recurrently track the desirable output $\tilde{y}$; ii) Design a state feedback switching signal $\sigma(t) = Wx(t)$, $W \in \mathcal{L}_{\mu \times 2^n}$ and a state feedback control $u(t) = \tilde{G}x(t)$, $\tilde{G} \in \mathcal{L}_{2^m \times 2^n}$ such that the outputs of system (2.13) can recurrently track $\tilde{y}$.

5.3 Recurrent Output Tracking of SBCNs

The ROTP is formally equivalent to determining whether the outputs of an SBCN intermittently and infinitely coincide with the desired output. This problem

is addressed through the concept of recurrent states, which is given as follows:

Definition 5.5. Given system (2.13) and a state $x = \delta_{2^n}^i \in \Delta_{2^n}$. $\delta_{2^n}^i$ is said to be a recurrent state of system (2.13), if there exist an integer $T \geq 0$, a switching signal sequence $\sigma = \{\sigma(0), \sigma(1), \dots, \sigma(T-1)\}$ and a control sequence $u = \{u(0), u(1), \dots, u(T-1)\}$, such that $x(T; \delta_{2^n}^i, \sigma, u) = \delta_{2^n}^i$.

From the above definition, we know a state that reaches itself under the σ and u will reach itself again by repeating the same switching and control sequences. Therefore, recurrent states are those that return to themselves infinitely often. If a state x is a recurrent state of system (2.13) and the output $y = Hx = \tilde{y}$, then the output trajectory generated by x will intermittently and infinitely reach the output $\tilde{y}$, that is it will recurrently track $\tilde{y}$. Thus, the set of recurrent states whose output equals $\tilde{y}$ is the required target state set, and the ROTP can be regarded as the case that all states of the system can reach this target set. Firstly, we determine the target state set.

Assume the desired output signal $\tilde{y} = \delta_{2^p}^\gamma$. Define the set $\Gamma(\tilde{y})$ as:

$$\Gamma(\tilde{y}) = \{\delta_{2^n}^i \mid \text{Col}_i(H) = \delta_{2^p}^\gamma, i \in \{1, 2, \dots, 2^n\}\}. \quad (5.4)$$

$\Gamma(\tilde{y})$ includes all states whose output equals $\delta_{2^p}^\gamma$. As mentioned in Chapter 4, we suppose $\Gamma(\tilde{y}) \neq \emptyset$ to make the ROTP meaningful.

Split the matrix L^q into 2^m blocks as $L^q = [L_1^q, L_2^q, \dots, L_{2^m}^q]$, where $L_k^q \in \mathcal{L}_{2^n \times 2^n}$, $q \in \{1, 2, \dots, \mu\}$ and $k \in \{1, 2, \dots, 2^m\}$. Define two matrices

$$\overline{M} := \sum_{\mathcal{B}} \sum_{k=1}^{2^m} \sum_{q=1}^{\mu} L_k^q \in \mathcal{B}_{2^n \times 2^n} \quad (5.5)$$

and

$$\overline{F} := \sum_{\mathcal{B}} \sum_{j=1}^{\mu 2^{m+n}} \overline{M}^{(j)} \in \mathcal{B}_{2^n \times 2^n}. \quad (5.6)$$

If there exist at least one switching signal $\sigma(t) = \delta_\mu^q \in \Delta_\mu$ and one control $u(t) = \delta_{2^m}^k \in \Delta_{2^m}$ such that $x(t+1) = \delta_{2^n}^j = L\sigma(t)u(t)x(t) = L\delta_\mu^q \delta_{2^m}^k \delta_{2^n}^i = L_k^q \delta_{2^n}^i$, then, $(\delta_{2^n}^j)^T \delta_{2^n}^i = (\delta_{2^n}^j)^T L_k^q \delta_{2^n}^i = (L_k^q)_{j,i} = 1$. Considering

$$\sum_{\mathcal{B}} \sum_{k=1}^{2^m} \sum_{q=1}^{\mu} (\delta_{2^n}^j)^T L_k^q \delta_{2^n}^i = (\delta_{2^n}^j)^T \sum_{\mathcal{B}} \sum_{k=1}^{2^m} \sum_{q=1}^{\mu} L_k^q = (\delta_{2^n}^j)^T \overline{M} \delta_{2^n}^i = (\overline{M})_{j,i},$$

we know that the matrix $\overline{M}$ includes all possible one-step state transition information. In other words, if there exist at least one switching signal and one control

such that the state $\delta_{2^n}^j$ is one-step reachable from the state $\delta_{2^n}^i$, then $(\overline{M})_{j,i} = 1$; otherwise, $(\overline{M})_{j,i} = 0$. Furthermore, $\overline{F}$ includes all possible state transition information. That is, $(\overline{F})_{j,i} = 1$ implies that $\delta_{2^n}^j$ is reachable from $\delta_{2^n}^i$ under at least one switching signal sequence and one control sequence. Since there are μ subnetworks, 2^m controls and 2^n states for system (2.13), the worst case is that one state reaches another state within $\mu 2^{m+n}$ steps [7].

Based on (5.4) and (5.6), we denote the set

$$\Lambda = \{\delta_{2^n}^i \mid (\overline{F})_{i,i} = 1, \delta_{2^n}^i \in \Gamma(\tilde{y})\}. \quad (5.7)$$

Obviously, Λ contains the recurrent states whose outputs equal $\delta_{2^p}^\gamma$, which is the required target set for the ROTP of system (2.13).

After determining the target set, the ROTP becomes to check whether all states of system (2.13) can reach the set Λ . To this end, we give the definition of set controllability for system (2.13).

Definition 5.6. Given a nonempty state set S . System (2.13) is said to be set controllable to S , if for any initial state $x_0 \in \Delta_{2^n}$, there exist an integer $T \geq 0$, a switching signal sequence $\sigma = \{\sigma(0), \sigma(1), \dots, \sigma(T-1)\}$ and a control sequence $u = \{u(0), u(1), \dots, u(T-1)\}$, such that $x(T; x_0, \sigma, u) \in S$.

Now, we discuss the condition for determining the set controllability of system (2.13) via the mode-dependent truth matrix method suggested in [97].

Given a switching signal $q \in \{1, 2, \dots, \mu\}$ and a nonempty set S , we denote by $T^q(S)$ a mode-dependent truth matrix whose rows and columns represent values of the control and state respectively:

$$(T^q(S))_{k,j} = \begin{cases} 1, & \text{if } L_k^q \delta_{2^n}^j \in S, \forall \delta_{2^n}^j \in \Delta_{2^n}, \\ 0, & \text{otherwise.} \end{cases} \quad (5.8)$$

When $(T^q(S))_{k,j} = 1$, the state $\delta_{2^n}^j$ can reach set S under the switching signal δ_μ^q and the control $\delta_{2^m}^k$ in one step; otherwise, it can not reach S in one step. Thus, $\text{Col}_j(T^q(S)) \neq \mathbf{0}_{2^m}$ implies that there exists at least one control such that the state $\delta_{2^n}^j$ can reach the set S under the switching signal δ_μ^q in one step. The rows where the non-zero position is located correspond to feasible controls.

Furthermore, we construct the l -th step reachable set of the set S as follows:

$$\Phi_l(S) = \bigcup_{q=1}^{\mu} \Phi_l^q(S) \setminus \bigcup_{i=1}^{l-1} \Phi_i(S), \quad l \in \mathbb{Z}_+, \quad (5.9)$$

where the set $\Phi_l^q(S) = \{\delta_{2^n}^j \mid \text{Col}_j(T_l^q(S)) \neq \mathbf{0}_{2^m}\}$, and

$$(T_l^q(S))_{k,j} = \begin{cases} 1, & \text{if } L_k^q \delta_{2^n}^j \in \Phi_{l-1}(S), \forall \delta_{2^n}^j \in \Delta_{2^n}, \\ 0, & \text{otherwise.} \end{cases} \quad (5.10)$$

Here, the set $\Phi_0(S) := S$. Apparently, $\bigcup_{i=1}^{l-1} \Phi_i(S) = \emptyset$ when $l = 1$ and $\Phi_{l_1}(S) \neq \Phi_{l_2}(S)$ when $l_1 \neq l_2$.

Lemma 5.7. *System (2.13) is set controllable to S , if and only if there exists an integer $\varepsilon > 0$, such that*

$$\bigcup_{l=1}^{\varepsilon} \Phi_l(S) = \Delta_{2^n}. \quad (5.11)$$

Proof: (Sufficiency) If there exists an integer $\varepsilon > 0$ such that (5.11) holds, then for every state $x = \delta_{2^n}^j \in \Delta_{2^n}$, there must be a unique integer $0 < \varepsilon' \leq \varepsilon$, such that $\delta_{2^n}^j \in \Phi_{\varepsilon'}(S)$. According to the construction of (5.9) and (5.10), we can find a switching signal $\delta_{\mu}^{q_j} \in \Delta_{\mu}$ and a control $\delta_{2^m}^{k_j} \in \Delta_{2^m}$, such that $\delta_{2^n}^j \in \Phi_{\varepsilon'}^{q_j}(S)$ and $L^{q_j} \delta_{2^m}^{k_j} \delta_{2^n}^j \in \Phi_{\varepsilon'-1}(S)$. Then, we choose $\sigma = W \delta_{2^n}^j = \delta_{\mu}^{q_j}$ and $u = \tilde{G} \delta_{2^n}^j = \delta_{2^m}^{k_j}$, where $W \in \mathcal{L}_{\mu \times 2^n}$, $\tilde{G} \in \mathcal{L}_{2^m \times 2^n}$ and $\delta_{2^m}^{k_j} \leq \text{Col}_j(T_{\varepsilon'}^{q_j}(S))$. Therefore, for any initial state $x(0) = \delta_{2^n}^j \in \Delta_{2^n}$, we can get a unique integer $0 < \varepsilon' \leq \varepsilon$, a state feedback switching signal $\sigma(t) = Wx(t)$, $W = \delta_{\mu}[q_1, q_2, \dots, q_{2^n}]$, and a state feedback control $u(t) = \tilde{G}x(t)$, $\tilde{G} = \delta_{2^m}[k_1, k_2, \dots, k_{2^n}]$, such that $x(\varepsilon'; x_0, \sigma, u) \in S$. According to Definition 5.6, system (2.13) is set controllable to S .

(Necessity) If system (2.13) is set controllable to S , then for any initial state $x_0 \in \Delta_{2^n}$, we can find an integer $\tau = \tau(x_0)$, a switching signal $\sigma = \{\sigma(0), \sigma(1), \dots, \sigma(\tau - 1)\} = \delta_{\mu}\{q_0, q_1, \dots, q_{\tau-1}\}$ and a control $u = \{u(0), u(1), \dots, u(\tau - 1)\} = \delta_{2^m}\{k_0, k_1, \dots, k_{\tau-1}\}$ such that

$$x(\tau; x_0, \sigma, u) \in S. \quad (5.12)$$

According to the state evolutionary dynamics and (5.12), we have $x(\tau) = L\sigma(\tau - 1)u(\tau - 1)x(\tau - 1) = L\delta_{\mu}^{q_{\tau-1}}\delta_{2^m}^{k_{\tau-1}}x(\tau - 1) \in S = \Phi_0(S)$. From (5.9), we know that $x(\tau - 1) \in \Phi_1(S)$. Furthermore, we have $x(\tau - 1) = L\sigma(\tau - 2)u(\tau - 2)x(\tau - 2) = L\delta_{\mu}^{q_{\tau-2}}\delta_{2^m}^{k_{\tau-2}}x(\tau - 2) \in \Phi_1(S)$, which implies that $x(\tau - 2) \in \Phi_2(S)$. If $\tau - 2 = 0$, then $x(0) = x_0 \in \Phi_2(S)$; otherwise, $x(\tau - 3) \in \Phi_3(S)$. Keep continuing the process, $x(0) \in \Phi_{\tau}(S)$. Thus, for any initial state $x_0 \in \Delta_{2^n}$, there exists the smallest integer

$0 < \tau(x_0) \leq \tau$, such that $x_0 \in \Phi_{\tau(x_0)}(S)$. Due to the arbitrariness of x_0 , let $\varepsilon = \max\{\tau(x_0) \mid x_0 \in \Delta_{2^n}\}$. Then, (5.11) holds for the integer ε . $\square$

We point out that the upper bound of ε is 2^n since there are 2^n states and the most extreme scenario is when each $\Phi_l(S)$ is a single point set.

The following theorem is provided to show the relationship between ROT and set controllability for system (2.13).

Theorem 5.8. *The ROTP of system (2.13) is solvable, if and only if system (2.13) is set controllable to the set $\Lambda \neq \emptyset$, where Λ is defined in (5.7).*

Proof: (Sufficiency) Condition $\Lambda \neq \emptyset$ implies that there exists at least one state $\delta_{2^n}^i$ in Λ such that $(\bar{F})_{i,i} = 1$ and $\delta_{2^n}^i \in \Gamma(\tilde{y})$, that is, there exist an integer $\tau' \geq 0$, a switching signal sequence $\sigma = \{\sigma(0), \sigma(1), \dots, \sigma(\tau' - 1)\}$ and a control sequence $u = \{u(0), u(1), \dots, u(\tau' - 1)\}$ such that

$$\begin{cases} x(\tau'; \delta_{2^n}^i, \sigma, u) = \delta_{2^n}^i, \\ H\delta_{2^n}^i = \tilde{y}. \end{cases}$$

Therefore, for any state $\delta_{2^n}^i \in \Lambda$ and any integer $t \in \mathbb{N}$, there exists an integer $\tilde{\tau} = \alpha \times \tau' \geq t$, where $\alpha \in \mathbb{Z}_+$, a switching signal sequence σ and a control sequence u such that $y(\tilde{\tau}; \delta_{2^n}^i, \sigma, u) = \tilde{y}$, which implies that the outputs generated from all states in Λ can recurrently track $\tilde{y}$.

If system (2.13) is set controllable to Λ , without loss of generality, for an initial state $x_0 = \delta_{2^n}^j$, we can find a proper state $\delta_{2^n}^i \in \Lambda$, a switching signal sequence $\sigma = \{\sigma(0), \sigma(1), \dots, \sigma(\bar{\tau} - 1)\}$ and a control sequence $u = \{u(0), u(1), \dots, u(\bar{\tau} - 1)\}$ such that

$$x(\bar{\tau}; \delta_{2^n}^j, \sigma, u) = \delta_{2^n}^i \in \Lambda. \quad (5.13)$$

Based on the analysis of the set Λ , we deduce that for the state $\delta_{2^n}^j$ and any integer $t \in \mathbb{N}$, there exists $\tau = \bar{\tau} + \tilde{\tau} \geq t$, $\sigma = \{\sigma(0), \sigma(1), \dots, \sigma(\tau - 1)\}$ and $u = \{u(0), u(1), \dots, u(\tau - 1)\}$ such that $y(\tau; \delta_{2^n}^j, \sigma, u) = \tilde{y}$. It implies that the outputs generated from state $\delta_{2^n}^j$ can recurrently track $\tilde{y}$. Considering the arbitrariness of x_0 , we conclude that the outputs of system (2.13) can recurrently track $\tilde{y}$.

(Necessity) We prove it by contradiction. If $\Lambda = \emptyset$, there are two cases. The first one is that $(\bar{F})_{i,i} = 0$ for any state $\delta_{2^n}^i \in \Gamma(\tilde{y})$, i.e., there is no recurrent state whose output equals $\tilde{y}$. The other case is that $\delta_{2^n}^i \notin \Gamma(\tilde{y})$ for any state $\delta_{2^n}^i$ satisfying $(\bar{F})_{i,i} = 1$, which implies that for any such $\delta_{2^n}^i$ and any $t \in \mathbb{Z}_+$, there is no integer $T > t$ and control sequence u such that $y(T; \delta_{2^n}^i, u) = \tilde{y}$.

If system (2.13) is not set controllable to the set Λ , according to the analysis of sufficiency, we know that at least one integer $j \in \{1, 2, \dots, 2^n\}$ does not satisfy (5.13), which means that for the state $\delta_{2^n}^j$, some time $\bar{t} \geq 0$, any σ and any u , we have $y(t; \delta_{2^n}^j, \sigma, u) \neq \tilde{y}$ for any $t > \bar{t}$.

Both two hypotheses contradict Definition 5.4. Hence, the conclusion holds. $\square$

According to Lemma 5.7 and Theorem 5.8, the following result is given to detect the solvability of the ROTP of (2.13).

Corollary 5.9. *The ROTP of system (2.13) is solvable if and only if the set $\Lambda \neq \emptyset$ and there exists an integer $\varepsilon > 0$ such that*

$$\bigcup_{l=1}^{\varepsilon} \Phi_l(\Lambda) = \Delta_{2^n}. \quad (5.14)$$

Note that if the ROTP is solvable under switching signal sequences and control sequences, there must exist the corresponding state feedback switching signals and the corresponding state feedback controls. The reason is as follows. The ROTP can be regarded as the set controllability problem. If there exist a switching signal sequence and a control sequence, it is possible to choose the shortest one. For an initial state $x_0 = \delta_{2^n}^{j_1}$, there exist the switching signal sequence $\sigma = \{\sigma(0), \sigma(1), \dots, \sigma(T-1)\} = \delta_\mu\{q_1, q_2, \dots, q_T\}$ and the control sequence $u = \{u(0), u(1), \dots, u(T-1)\} = \delta_{2^m}\{k_1, k_2, \dots, k_T\}$, such that $x(t; x_0, \sigma, u) \in \Lambda$. We assume the generated state trajectory is $\delta_{2^n}^{j_1} \rightarrow \delta_{2^n}^{j_2} \rightarrow \dots \rightarrow \delta_{2^n}^{j_T}$. Then let $\text{Col}_{j_i}(W) = \delta_\mu^{q_i}$ and $\text{Col}_{j_i}(\tilde{G}) = \delta_{2^n}^{k_i}$. We collect one state from $\Delta_{2^n} \setminus \{\delta_{2^n}^{j_1}, \delta_{2^n}^{j_2}, \dots, \delta_{2^n}^{j_{T-1}}\}$, do the same process, then obtain some other columns of W and $\tilde{G}$. Until all the states are checked from generated state trajectories, all columns of W and $\tilde{G}$ can be determined.

In what follows, based on the analysis of Theorem 5.8 and (5.7), we propose Algorithm 8 to find the matrices W and $\tilde{G}$ such that the output of (2.13) can recurrently track the signal $\tilde{y}$.

The first step of Algorithm 8 ensures that there exist switching signal sequences and control sequences, such that some states can return themselves and their outputs equal $\tilde{y}$. Assume that there are α elements in Λ and denote it by $\Lambda = \{\delta_{2^n}^{j_1}, \delta_{2^n}^{j_2}, \dots, \delta_{2^n}^{j_\alpha}\}$. In the following procedures, for each state $\delta_{2^n}^{j_i}$, we can find an integer ε'_i , such that $\delta_{2^n}^{j_i} \in \Phi_{\varepsilon'_i}(\Lambda)$. We then deduce that all states in Λ can reach the set Λ . From each $\Phi_{\varepsilon'_i}(\Lambda)$, according to the corresponding mode-dependent truth matrices, the $\tilde{G}$ and W satisfying all states in Λ reach Λ can be designed. When all states in $\Delta_{2^n} \setminus \bigcup_{i=1}^{\alpha} \Phi_{\varepsilon'_i}(\Lambda)$ have been checked, the $\tilde{G}$ and W such that the ROTP is

Algorithm 8 Switching signal and control design for the ROTP of system (2.13)

Require: Matrix L
Ensure: Matrices W and $\tilde{G}$

- 1: Calculate the set Λ according to equations (5.5), (5.6), and (5.7).
 - 2: **for** each state $\delta_{2^n}^j \in \Delta_{2^n}$ **do**
 - 3: Find the unique integer ε' ($0 < \varepsilon' \leq \varepsilon$) such that $\delta_{2^n}^j \in \Phi_{\varepsilon'}(\Lambda)$.
 - 4: Find a switching signal $\delta_\mu^{q_j} \in \Delta_\mu$ and a control $\delta_{2^m}^{k_j} \in \Delta_{2^m}$ satisfying $\delta_{2^n}^j \in \Phi_{\varepsilon'}^{q_j}(\Lambda)$ and $L^{q_j} \delta_{2^m}^{k_j} \delta_{2^n}^j \in \Phi_{\varepsilon'-1}(\Lambda)$.
 - 5: Set $\text{Col}_j(W) \leftarrow \delta_\mu^{q_j}$ and $\text{Col}_j(\tilde{G}) \leftarrow \delta_{2^m}^{k_j}$.
 - 6: **end for**
 - 7: **return** $W = \delta_\mu[q_1, q_2, \dots, q_{2^n}]$, and $\tilde{G} = \delta_{2^m}[k_1, k_2, \dots, k_{2^n}]$.
-

solvable can be designed.

Remark 5.10. In Algorithm 8, if all states in the set Λ reach itself by one step or they form some cycles, then the ROTP becomes the traditional OTP for system (2.13).

5.4 An Example

A biological example is provided in this section to test the obtained results.

Example 5.11. Consider a modified model based on the lac operon in the bacterium *Escherichia coli* [129], whose dynamics are

$$\begin{cases} X_1(t+1) = f_1^{\Sigma(t)}(U_1(t), U_2(t), X_1(t), X_2(t), X_3(t)), \\ X_2(t+1) = f_2^{\Sigma(t)}(U_1(t), U_2(t), X_1(t), X_2(t), X_3(t)), \\ X_3(t+1) = f_3^{\Sigma(t)}(U_1(t), U_2(t), X_1(t), X_2(t), X_3(t)), \end{cases} \quad (5.15)$$

where $\Sigma(t) \in \{1, 2\}$ and

$$\begin{cases} f_1^1 = X_3(t) \vee X_2(t), \\ f_2^1 = X_1(t) \wedge U_1(t), \\ f_3^1 = X_3(t), \end{cases} \quad (5.16)$$

and

$$\begin{cases} f_1^2 = X_3(t) \vee X_2(t), \\ f_2^2 = X_1(t) \wedge U_1(t), \\ f_3^2 = X_1(t) \wedge U_2(t) \vee U_1(t). \end{cases} \quad (5.17)$$

Assume the output dynamics of system (5.15) are

$$\begin{cases} Y_1(t) = X_1(t) \wedge \neg(X_2(t) \wedge X_3(t)), \\ Y_2(t) = \neg X_3(t). \end{cases} \quad (5.18)$$

The algebraic forms of system (5.15) and (5.18) can be expressed as

$$\begin{cases} x(t+1) = L\sigma(t)u(t)x(t), \\ y(t) = Hx(t), \end{cases} \quad (5.19)$$

where

$$\begin{aligned} L_1 &= \delta_8[1, 2, 1, 6, 3, 4, 3, 8, 1, 2, 1, 6, 3, 4, 3, 8, 3, 4, 3, 8, 3, 4, 3, 8, 3, 4, 3, 8, 3, 4, 3, 8], \\ L_2 &= \delta_8[1, 1, 1, 5, 3, 3, 3, 7, 1, 1, 1, 5, 3, 3, 3, 7, 3, 3, 3, 7, 4, 4, 4, 8, 4, 4, 4, 8, 4, 4, 4, 8], \\ H &= \delta_4[2, 1, 4, 1, 4, 3, 4, 3]. \end{aligned}$$

Assume that the reference signal is $\tilde{y} = \delta_4^1$.

From the output matrix H , we can easily see that $\Gamma(\delta_4^1) = \{\delta_8^2, \delta_8^4\}$. After a simple calculation, we get matrices $\overline{M}$ and $\overline{F}$. And based on (5.7), we have $\Lambda = \{\delta_8^2, \delta_8^4\} \neq \emptyset$.

Denote $\Phi_0 = \Lambda$. According to 5.9 and 5.10, we evaluate the mode-dependent truth matrices

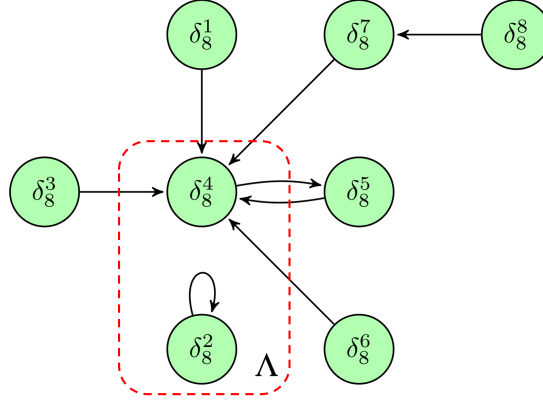
$$T_1^1(\Lambda) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \in \mathcal{B}_{4 \times 8},$$

and

$$T_1^2(\Lambda) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \in \mathcal{B}_{4 \times 8}.$$

Then, $\Phi_1^1(\Lambda) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^5, \delta_8^6, \delta_8^7\}$ and $\Phi_1^2(\Lambda) = \{\delta_8^2, \delta_8^6\}$. Thus,

$$\Phi_1(\Lambda) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^5, \delta_8^6, \delta_8^7\}.$$


 Figure 5.1: The state transition diagram under W and $\tilde{G}$

We evaluate matrices $T_2^1(\Lambda)$ and $T_2^2(\Lambda)$

$$T_2^1(\Lambda) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathcal{B}_{4 \times 8},$$

$$T_2^2(\Lambda) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \in \mathcal{B}_{4 \times 8}.$$

Then, $\Phi_1^1(\Lambda) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^4, \delta_8^5, \delta_8^6, \delta_8^7, \delta_8^8\}$ and $\Phi_2^2(\Lambda) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^4, \delta_8^5, \delta_8^7\}$. We obtain that $\Phi_2(\Lambda) = \{\delta_8^4, \delta_8^8\}$.

Since $\Lambda \neq \emptyset$ and $\Phi_1(\Lambda) \cup \Phi_2(\Lambda) = \Delta_{2^n}$, we conclude that the ROTP of system (5.19) is solvable. Further, we design the matrices W and $\tilde{G}$.

For the state $\delta_8^1 \in \Phi_1(\Lambda)$, we have $\sigma(t) = \delta_2^1$. Then $u(t) = \tilde{G}x(t) = \delta_4^1$ satisfies $\text{Col}_1(\tilde{G}) \leq \text{Col}_1(T_1^1(\Lambda))$. For the state $\delta_8^4 \in \Phi_2(\Lambda)$, we have $\sigma(t) = \delta_2^1$ or δ_2^2 . If $\sigma(t) = \delta_2^1$, then $u(t) = \tilde{G}x(t) = \delta_4^1, \delta_4^2$, or δ_4^3 , which satisfies $\text{Col}_4(\tilde{G}) \leq \text{Col}_4(T_2^1(\Lambda))$. If $\sigma(t) = \delta_2^2$, then $u(t) = \tilde{G}x(t) = \delta_4^1$, or δ_4^2 such that $\text{Col}_4(\tilde{G}) \leq \text{Col}_4(T_2^2(\Lambda))$ holds. Analogously, one can obtain the results for remaining states.

Finally, according to Algorithm 8, there are 900 different combinations of W and $\tilde{G}$. Here, we list one of them, that is, $W = \delta_2[1, 2, 1, 1, 1, 2, 1, 1]$ and $\tilde{G} = \delta_4[4, 1, 4, 1, 3, 1, 3, 1]$. Figures 5.1 and 5.2 show the corresponding state transition diagram and the output trajectory respectively.

Note that in this case, the state δ_8^2 can reach itself every time and its output

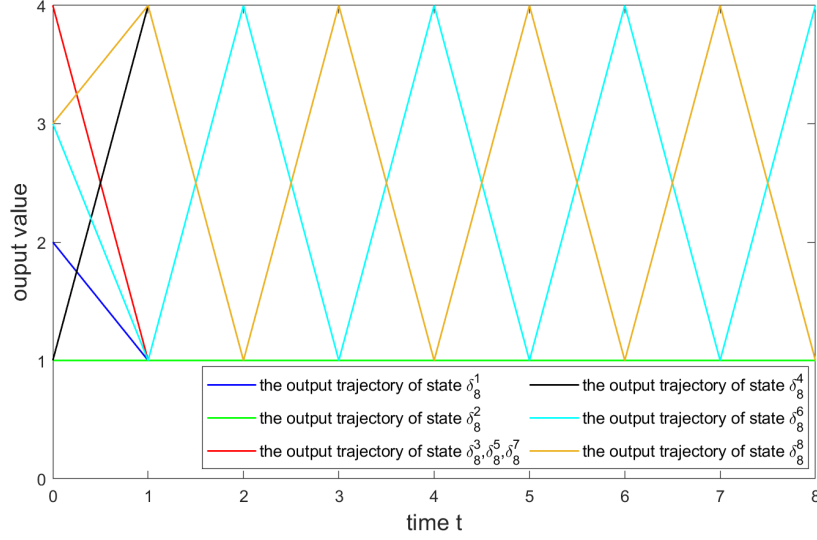


Figure 5.2: The output trajectories of system (5.19) under W and $\tilde{G}$

equals $\tilde{y}$. If all states of the system reach to δ_8^2 , then the system can achieve the traditional output tracking. However, only one state can reach δ_8^2 . Except for δ_8^2 , other states can enter the recurrent state trajectory $\delta_8^4 \rightarrow \delta_8^5 \rightarrow \delta_8^4$. And the corresponding output trajectory is $\delta_4^1 \rightarrow \delta_4^4 \rightarrow \delta_4^1$. Overall, although it can not achieve the traditional OT, the system can recurrently track the signal $\tilde{y}$.

5.5 Conclusion

This chapter addressed the ROTP in SBCNs, introducing its definition and a solvability condition based on recurrent states and reachable sets. State feedback controllers and switching signals were designed to achieve ROT. Finally, the validity of obtained results were verified through a practical example.

Chapter 6

Strategy Optimization of Controlled Heterogeneous Hyper-networked Evolutionary Games

6.1 Introduction

In the previous chapters, the analysis and control problems of logical dynamical systems have been systematically investigated, including identification and recurrent output tracking for B(C)Ns and SB(C)Ns. Furthermore, NEGs with finite strategy sets can also be viewed as a class of logical dynamical systems. This connection enables the algebraic framework developed in the previous chapters to be naturally extended to the analysis and control of evolutionary game dynamics.

As an important application of logical dynamical systems, NEGs [14] provide a powerful framework for modeling strategic interactions among multiple players in complex real-world systems, where collective behavior emerges from individual decision-making processes. However, most existing works on NEGs are based on pairwise interactions, which are insufficient to capture higher-order interactions commonly observed in real-world scenarios, such as group cooperation, multi-party competition, and social dynamics [17, 18, 19, 20]. To overcome this limitation, HNEGs have been introduced to model multi-player interactions [22]. Nevertheless, existing studies mainly focus on homogeneous hypergraphs or simplified interaction structures [115, 116, 120], and the strategy optimization problem under heterogeneous

hypergraphs with payoff constraints remains largely unexplored.

Motivated by this, this chapter investigates the strategy optimization problem of heterogeneous HNEGs. By employing the STP method, a unified algebraic representation of heterogeneous HNEGs is established to represent the evolutionary dynamics with hyperedge-induced multi-player payoff structures. By introducing control players as external inputs, the strategy optimization problem is reformulated as a minimum-time set controllability problem. Based on this formulation, necessary and sufficient conditions are derived to guarantee that the payoff of each player reaches and remains above a prescribed threshold, and corresponding state feedback control methods are developed to achieve this objective.

6.2 Algebraic Formulation of Heterogeneous HNEGs

In this section, we establish an algebraic framework for heterogeneous HNEGs (2.11) introduced in Chapter 2. By introducing payoff matrices associated with hyperedges of different orders, the overall payoff structure is unified into an algebraic form based on the STP method, which facilitates the formulation of the evolutionary dynamics.

We first assume that all players share a common finite strategy set, i.e., $\mathcal{S}_i = \{1, 2, \dots, k\}$, $\forall i \in \{1, 2, \dots, n\}$. And we adopt the widely used *myopic best response adjustment rule* as the strategy update rule, which means that at each time step, each player selects the strategy that maximizes its instantaneous payoff given the current strategies of all other players.

In a heterogeneous HNEG, player i interacts with its neighbors through higher-order interactions and its overall payoff at time t can be expressed as

$$c_i(t) = \sum_{l=1}^{k_i} c_i^{E_i^l}(x_i(t), x_{j_1}(t), x_{j_2}(t), \dots, x_{j_{g_i^l-1}}(t)), \quad (6.1)$$

where $c_i^{E_i^l}$ represents the payoff function of player i on the hyperedge E_i^l and $g_i^l := |E_i^l|$. Note that although the same notation c_i is retained, it now represents the overall payoff of player i at time t in the entire hyper-network, rather than the payoff function of a finite normal game in Definition 2.7.

Example 6.1. Recall Example 2.10. For player 3 in Figure 2.1, we have $E(3) = \{E_3^1, E_3^2, E_3^3\}$. The corresponding neighbor sets on each hyperedge are $N(3, E_3^1) = \{2, 3\}$, $N(3, E_3^2) = \{1, 3, 4\}$, and $N(3, E_3^3) = \{3, 4, 5\}$, respectively. Consequently,

its overall payoff at time t can be written as

$$c_3(t) = c_3^{E_1^1}(x_3(t), x_2(t)) + c_3^{E_2^2}(x_3(t), x_1(t), x_4(t)) + c_3^{E_3^3}(x_3(t), x_4(t), x_5(t)).$$

To derive an algebraic representation of the payoff functions, we introduce payoff matrices associated with hyperedges of different orders. In classical 2-player FNGs, the payoff structure can be represented by a matrix, as illustrated in Figure 6.1a), where the row and column indices correspond to the strategies of the two players, respectively. For instance, if player 1 and player 2 adopt strategies C and D , respectively, then player 1 obtains payoff S while player 2 obtain payoff T . However, in FNGs arising in heterogeneous HNEGs, the payoff structure may no longer be described by a two-dimensional matrix. For example, in a 3-player FNG, the payoff depends on the strategies of three players and can be naturally visualized as a three-dimensional payoff cube, as illustrated in Figure 6.1b), whose three axes correspond to the strategies of the three players. If player 1, player 2 and player 3 adopt strategies C , D and D , respectively, then their corresponding payoffs are S , W and W .

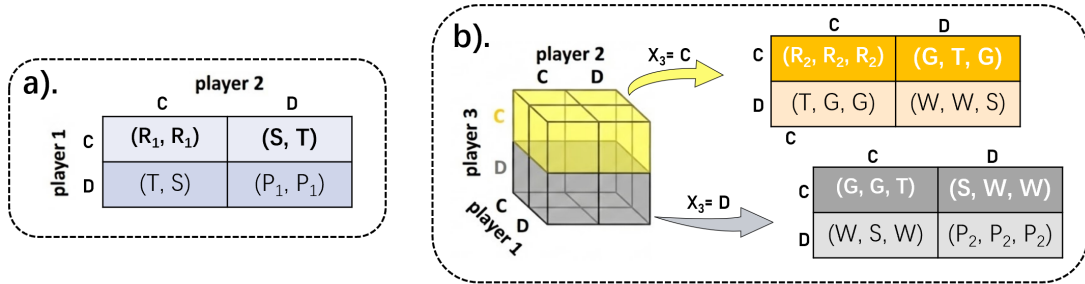


Figure 6.1: Payoff structures associated with different FNGs. a). The payoff structure of a 2-player FNG. b). The payoff structure of a 3-player FNG.

We adopt a vectorization-based matrix formulation [107] to obtain a unified algebraic representation for FNGs $G_1, G_2, \dots, G_\alpha$. For a hyperedge E^l involving g_l players, the payoff of each player is represented by a payoff matrix obtained by enumerating all possible strategy profiles, as summarized in Table 6.1. The symbols “Pro” and “P” in the table represent profile and player respectively.

Remark 6.2. *Since the hypergraph considered in this chapter is undirected, the players contained in each hyperedge have no inherent ordering. Consequently, for any hyperedge containing players $i_1, i_2, i_3 \in N$, all permutations such as $(i_1, i_2, i_3), (i_1, i_3, i_2), (i_2, i_1, i_3), (i_2, i_3, i_1), (i_3, i_1, i_2)$ and (i_3, i_2, i_1) represent the same hyperedge. For notational convenience and without loss of generality, we sort player indices in ascending*

Table 6.1: Payoff matrix associated with a hyperedge E^l involving g_l players

P \ Pro	11... 1	11... 2	...	11... k	...	kk... k
i_1	a_{11}	a_{12}	$\dots$	a_{1k}	$\dots$	$a_{1k^{g_l}}$
i_2	a_{21}	a_{22}	$\dots$	a_{2k}	$\dots$	$a_{2k^{g_l}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i_{g_l}	$a_{g_l 1}$	$a_{g_l 2}$	$\dots$	$a_{g_l k}$	$\dots$	$a_{g_l k^{g_l}}$

order throughout the chapter and consider only representations of the form (i_1, i_2, i_3) with $i_1 < i_2 < i_3$.

Let

$$V_q^{g_l} := [a_{q1}, a_{q2}, \dots, a_{qk^{g_l}}],$$

where $q \in \{1, 2, \dots, g_l\}$. Each entry of $V_q^{g_l}$ corresponds to the payoff received by the q -th player under one of the k^{g_l} possible profiles on the hyperedge E^l . We identify each strategy set $\mathcal{S}_i$, $i \in N$, with the logical vector space $\mathcal{S}_i \sim \Delta_k$, so that the strategy $l \in \mathcal{S}_i$ is written as the vector form δ_k^l . Thus, using the STP method and Lemma 2.5, the algebraic form of the payoff of player i on the hyperedge $E_i^l \in E(i)$ can be written as

$$\begin{aligned}
& c_i^{E_i^l}(x_i(t), x_{j_1}(t), \dots, x_{j_{g_l^i-1}}(t)) \\
&= c_i^{E_i^l}(x_{j_1}(t), \dots, x_{j_{q-1}}(t), x_i(t), x_{j_{q+1}}(t), \dots, x_{j_{g_l^i-1}}(t)) \\
&= V_q^{g_l^i} x_{j_1}(t) \cdots x_{j_{q-1}}(t) x_i(t) x_{j_{q+1}}(t) \cdots x_{j_{g_l^i-1}}(t) \\
&= V_q^{g_l^i} D_r^{[k^{j_1-1}, k]} D_f^{[k^{j_1}, k^{j_2-j_1-1}]} \cdots D_f^{[k^{j_{q-1}}, k^{j_q-j_{q-1}-1}]} D_f^{[k^i, k^{j_{q+1}-i-1}]} \cdots D_f^{[k^{j_{g_l^i-1}}, k^{n-j_{g_l^i-1}-1}]} x(t) \\
&:= W_i^{g_l^i} x(t), \tag{6.2}
\end{aligned}$$

where $W_i^{g_l^i} \in \mathbb{R}^{k^n}$ is the corresponding structure row vector, and $x(t) := \times_{j=1}^n x_j(t)$ denotes the strategy profile of all players. It is worth noting that $W_i^{g_l^i}$ admits a unified expression for $1 < q < g_l^i$. The boundary cases $q = 1$ (when player i is the first on the hyperedge) and $q = g_l^i$ (when player i is the last on the hyperedge) correspond to empty left/right segments, and are therefore written separately as

$$W_i^{g_l^i} = \begin{cases} V_1^{g_l^i} D_r^{[k^{i-1}, k]} D_f^{[k^i, k^{j_1-i-1}]} \cdots D_f^{[k^{j_{g_l^i-1}}, k^{n-j_{g_l^i-1}-1}]} & q = 1, \\ V_{g_l^i}^{g_l^i} D_r^{[k^{j_1-1}, k]} D_f^{[k^{j_1}, k^{j_2-j_1-1}]} \cdots D_f^{[k^{i-1}, k^{n-i}]} & q = g_l^i. \end{cases}$$

The total payoff of player i , obtained from all k_i hyperedges to which it belongs, can be expressed in the following algebraic form:

$$c_i(t) = \sum_{l=1}^{k_i} W_i^{g_l^i} x(t) := \widetilde{W}_i x(t), \quad (6.3)$$

where $\widetilde{W}_i$ is the structure matrix associated with the overall payoff function c_i , which shows the payoff information of the player i in all possible strategy profiles.

Example 6.3. Recall Example 2.10. Consider an HNEG defined on hypergraph illustrated in Figure 2.1. Each player has two available strategies 1 and 2. There are a 2-player FNG G_1 and a 3-player FNG G_2 . Assume that the payoff matrices of G_1 and G_2 are given in Tables 6.2 and 6.3, respectively. According to the above

Table 6.2: Payoff matrix on G_1

P \ Pr	11	12	21	22
i_1	0	1	-1	0
i_2	0	-1	1	0

Table 6.3: Payoff matrix on G_2

P \ Pr	111	112	121	122	211	212	221	222
i_1	0	1	1	-2	-2	1	1	0
i_2	0	1	-2	1	1	-2	1	0
i_3	0	-2	1	1	1	1	-2	0

two tables, we know that $V_1^2 = [0, 1, -1, 0]$, $V_2^2 = [0, -1, 1, 0]$, $V_1^3 = [0, 1, 1, -2, -2, 1, 1, 0]$, $V_2^3 = [0, 1, -2, 1, 1, -2, 1, 0]$, and $V_3^3 = [0, -2, 1, 1, 1, 1, -2, 0]$.

Consider the player 3, whose three incident hyperedges are $E(3) = \{E_3^1, E_3^2, E_3^3\}$. According to equations (6.2) and (6.3), we can calculate the player's overall payoff as follows.

$$\begin{aligned}
c_3(t) &= c_3^{E_3^1}(x_3(t), x_2(t)) + c_3^{E_3^2}(x_3(t), x_1(t), x_4(t)) + c_3^{E_3^3}(x_3(t), x_4(t), x_5(t)) \\
&= V_2^2 x_2(t) x_3(t) + V_2^3 x_1(t) x_3(t) x_4(t) + V_1^3 x_3(t) x_4(t) x_5(t) \\
&= V_2^2 D_r^{[2,2]} D_f^{[8,4]} x(t) + V_2^3 D_r^{[2,2]} D_f^{[16,2]} x(t) + V_1^3 D_r^{[4,2]} x(t) \\
&= (V_2^2 D_r^{[2,2]} D_f^{[8,4]} + V_2^3 D_r^{[2,2]} D_f^{[16,2]} + V_1^3 D_r^{[4,2]}) x(t) \\
&:= \widetilde{W}_3 x(t),
\end{aligned} \quad (6.4)$$

where

$$\begin{aligned} \widetilde{W}_3 = [0, -2, 2, 2, -2, -2, -2, 0, 2, 0, 0, 0, 2, 2, -2, 0, \\ 0, -2, 2, 2, -2, -2, -2, 0, 2, 0, 0, 0, 2, 2, -2, 0]. \end{aligned}$$

We now proceed to the mechanism governing strategy evolution in the heterogeneous HNEG. In the chapter, players revise their strategies following the *myopic best response adjustment rule*. Formally, the myopic best response of player i is defined as

$$x_i(t+1) \in Q_i := \arg \max_{x_i \in \mathcal{S}_i} c_i(x_i, x_{i'}(t)), \quad (6.5)$$

where $x_{i'}(t) := \times_{j \in N_i \setminus \{i\}} x_j(t)$.

When multiple strategies yield the same maximal payoff, a fixed priority rule is imposed to guarantee the uniqueness of the selected strategy; specifically, the strategy with the lowest priority index is chosen. More precisely, when $|Q_i| > 1$, the next strategy is determined by

$$x_i(t+1) = \min\{x \mid x \in Q_i\}. \quad (6.6)$$

To systematically determine the myopic best response for each player, we proceed as follows. First, after straightforward algebraic manipulations, equation (6.3) can be equivalently rewritten as

$$c_i(t) = \widetilde{W}_i x(t) = \widetilde{W}_i W_{[k, k^{i-1}]} x_i x_{-i}(t) := \overline{W}_i x_i x_{-i}(t), \quad (6.7)$$

where $x_{-i}(t) := \times_{j \neq i} x_j(t)$ denotes the strategy profile of all players except player i . We partition the row vector $\overline{W}_i$ into k blocks according to the number of strategies of player i :

$$\overline{W}_i := [\overline{W}_i^1, \overline{W}_i^2, \dots, \overline{W}_i^k],$$

where $\overline{W}_i^a \in \mathbb{R}^{k^{n-1}}$ for $a = 1, 2, \dots, k$, each of which contains all possible payoffs of player i when its own strategy is fixed at a , while the strategy profiles of all other players vary. Then, the best strategy of player i is determined by comparing the corresponding columns of the k submatrices. Specifically, for each possible $x_{-i}(t)$, corresponding to a column index $l \in \{1, 2, \dots, k^{n-1}\}$, we determine

$$a_i^* := \arg \max_{a \in \{1, 2, \dots, k\}} \text{Col}_l(\overline{W}_i^a).$$

If a_i^* is not unique, a unique value is selected according to the priority rule given in (6.6). Collecting the best responses obtained above, we construct the logical matrix

$\bar{L}_i := \delta_k[a_1^*, a_2^*, \dots, a_{k^{n-1}}^*] \in \mathcal{L}_{k \times k^{n-1}}$. This matrix maps $x_{-i}(t)$ to the basis vector corresponding to the best response of player i . Hence, the algebraic form of the strategy evolutionary dynamics for player i is obtained as

$$x_i(t+1) = \bar{L}_i x_{-i}(t) = \bar{L}_i D_f^{[k^{i-1}, k]} x(t) := L_i x(t), \quad (6.8)$$

where $L_i \in \mathcal{L}_{k \times k^n}$ is the structure matrix of the evolutionary dynamics of player i . Finally, multiplying the equations of all players together yields the algebraic form of profile dynamics of heterogeneous HNEGs

$$x(t+1) = \bar{L}x(t),$$

where $\bar{L} = L_1 * L_2 * \dots * L_n \in \mathcal{L}_{k^n \times k^n}$.

Example 6.4. Recall Example 6.3. We now calculate the algebraic form of evolutionary dynamics for player 3.

According to equation (6.7), we obtain

$$\begin{aligned} \bar{W}_3 = \widetilde{W}_3 W_{[4,2]} = [0, -2, 2, 2, 2, 0, 0, 0, 0, -2, 2, 2, 2, 0, 0, 0, -2, -2, \\ -2, 0, 2, 2, -2, 0, -2, -2, -2, 0, 2, 2, -2, 0]. \end{aligned}$$

Split the matrix $\bar{W}_3$ into 2 blocks, $\bar{W}_3^1 = [0, -2, 2, 2, 2, 0, 0, 0, 0, -2, 2, 2, 2, 0, 0, 0]$ and $\bar{W}_3^2 = [-2, -2, -2, 0, 2, 2, -2, 0, -2, -2, -2, 0, 2, 2, -2, 0]$. Since

$$\text{Col}_1(\bar{W}_3^1) > \text{Col}_1(\bar{W}_3^2),$$

we have $a_1^* = 1$. By comparing the remaining columns in the same manner, we obtain $\bar{L}_3 = \delta_2[a_1^*, a_2^*, \dots, a_{16}^*] = \delta_2[1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1]$. Therefore, the algebraic form of the evolutionary dynamics for player 3 is:

$$x_3(t+1) = \bar{L}_3 x_{-3}(t) = \bar{L}_3 D_f^{[4,2]} x(t) := L_3 x(t),$$

where $L_3 = \delta_2[1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1]$.

Based on the obtained algebraic representation, the evolutionary dynamics of the heterogeneous HNEG can be systematically analyzed, including the evolution of players' strategies and the resulting payoff profiles. However, under natural evolution, the system may converge to certain strategy profiles in which the payoffs of some players fall below the minimal level required for sustainable participation. In

practical scenarios, such situations may correspond to states where players cannot maintain viable operation or survival, and thus external intervention becomes necessary. To address this issue, the following subsection introduces control into the heterogeneous HNEG framework and aims to design proper control strategies such that the payoffs of all players exceed a prescribed threshold in minimum time and remain above this level thereafter.

6.3 Strategy Optimization

By embedding control players, whose strategies can be freely selected as control inputs, into the existing hyper-network structure, the original heterogeneous HNEG is transformed into a controlled heterogeneous HNEG.

Definition 6.5. [107] The triple $((N, E), G, \Pi)$ is called a controlled heterogeneous HNEG if the player set N is decomposed into two disjoint subsets $N = U \cup Z$ and $U \cap Z = \emptyset$, and for every $u \in U$, its strategy can be arbitrarily selected from its strategy set at each instant t . Here, $u \in U$ is called a control player and $z \in Z$ is called a state player.

We assume without loss of generality that the first v players in the set N act as control players, that is, $U = \{1, 2, \dots, v\}$, and $Z = \{v+1, v+2, \dots, n\}$. For all state players in the set Z , we still adopt the myopic best response adjustment rule as the strategy update rule. Under this assumption, equation (6.8) can be rewritten as

$$x_i(t+1) = L_i u(t) z(t), \quad (6.9)$$

where $u(t) := \times_{j=1}^v x_j(t)$ and $z(t) := \times_{j=v+1}^n x_j(t)$. By multiplying the equations of all state players, we obtain the following system dynamics:

$$z(t+1) = L u(t) z(t), \quad (6.10)$$

where $L = L_{v+1} * L_{v+2} * \dots * L_n \in \mathcal{L}_{k^{n-v} \times k^n}$ is the structure matrix of the controlled heterogeneous HNEG.

Let $\tilde{P}$ denote the prescribed payoff threshold for all players. Based on equation (6.3), we define the set Ω^i as follows:

$$\Omega^i = \{\delta_{k^n}^j \mid \text{Col}_j(\tilde{W}_i) \geq \tilde{P}\}. \quad (6.11)$$

This set consists of all strategy profiles such that the payoff of the i -th player reaches

or exceeds the prescribed threshold. The same procedure can be applied to derive the payoff information of all other players and their corresponding sets Ω^i . We further define the set

$$\Omega := \Omega^1 \cap \Omega^2 \cap \cdots \cap \Omega^n. \quad (6.12)$$

It follows that any strategy profile in Ω guarantees that all players' payoffs reach or exceed the prescribed threshold.

The strategy optimization problem is to determine a control sequence that drives the system into the set Ω in the shortest possible time and keeps it inside thereafter. This problem can be formulated as the minimum-time reach-and-stay optimal control problem with respect to the set Ω :

$$\begin{aligned} \min_u \quad & J(u, z_0) = T, \\ \text{s.t.} \quad & \begin{cases} z(t+1) = Lu(t)z(t) \\ z(t) \in \Delta_{k^{n-v}}, u(t) \in \Delta_{k^v} \\ z(0) = z_0 \\ u(t)z(t) \in \Omega, \forall t \geq T \end{cases} \end{aligned}$$

where $u = \{u(0), u(1), \dots\}$ represents a control sequence and T is an unknown but finite terminal time.

To solve the problem, we first guarantee the system stay in it once entering the set Ω . Since the control input can be freely selected, the invariant analysis is carried out on the state space $\Delta_{k^{n-v}}$ rather than on the profile space Δ_{k^n} . Thus, we characterize a maximum control invariant subset generated by set Ω , whose definition is introduced as follows.

Definition 6.6. [124] A set $\Phi \subseteq \Delta_{k^{n-v}}$ is called the control invariant subset generated by set Ω , if for any state $z(0) \in \Phi$, there exists a proper control sequence $\{u(0), u(1), \dots\}$ such that $u(t)z(t) \in \Omega$ for any $t \geq 0$. Φ^* is called the maximum control invariant subset generated by Ω if each control invariant subset generated by Ω is the subset of Φ^* .

Since $\Delta_{k^{n-v}}$ is a finite set and the admissible control inputs satisfy $u(t) \in \Delta_{k^v}$, the existence of a solution of the optimal problem can be guaranteed once feasibility is ensured. If the maximum control invariant subset Φ^* is nonempty, then the optimization problem can be viewed as a minimum-time set controllability problem, which is to design the strategies for control players such that any initial state $z(t)$ reach the set Φ^* in minimum time. In the following, we propose the following

proposition to clarify the relationship between strategy optimaization problem and minimum-time set controllability problem for system (6.10).

Proposition 6.7. *Consider system (6.10) and assume $\Phi^* \neq \emptyset$. For any initial state $z(0) := z_0 \in \Delta_{k^{n-v}}$, there exists a smallest integer $T > 0$ and a control sequence $\{u(0), u(1), \dots\}$ such that $u(t)z(t) \in \Omega$ for any $t \geq T$ if and only if for any initial state $z_0 \in \Delta_{k^{n-v}}$, there exists a smallest $T \geq 0$ and a control sequence $\{u(0), u(1), \dots\}$ such that $z(T) \in \Phi^*$.*

Proof. (Sufficiency) Suppose that for any initial state $z_0 \in \Delta_{k^{n-v}}$, there exist a smallest integer $T \geq 0$ and a control sequence $\{u(0), u(1), \dots\}$ such that $z(t) \in \Phi^*$, $\forall t \geq T$. Due to $z(T) \in \Phi^*$, we know from Definition 4.2 that $u(t)z(t) \in \Omega$, $\forall t \geq T$. Since T is the shortest time such that $z(T) \in \Phi^*$, we conclude that for any initial state $z_0 \in \Delta_{k^{n-v}}$, there exists a smallest integer $T > 0$ and a control sequence $\{u(0), u(1), \dots\}$ such that $u(t)z(t) \in \Omega$ for any $t \geq T$.

(Necessity) Assume that for any initial state $z_0 \in \Delta_{k^{n-v}}$, there exists a smallest integer $T > 0$ and a control sequence $\{u(0), u(1), \dots\}$ such that $u(t)z(t) \in \Omega$ for any $t \geq T$. Fix an arbitrary initial state z_0 . Consider the state $z(T)$. The sequence $\{u(T), u(T+1), \dots\}$ keeps the strategy profile in Ω for all future times. Hence, by Definition 4.2, it follows that $z(T) \in \Phi^*$.

We prove by contradiction that T is the smallest integer. If there exist an integer $\bar{T} < T$ such that $z(\bar{T}) \in \Phi^*$, then we have $u(t)z(t) \in \Omega$ for any $t \geq \bar{T}$, which contradicts the assumptions. Thus, for any initial state $z_0 \in \Delta_{k^{n-v}}$, there exist a smallest integer $T \geq 0$ and a control sequence $\{u(0), u(1), \dots\}$ such that $z(t) \in \Phi^*$, $\forall t \geq T$. $\square$

In the following, we first check the existence of a control sequence that ensures all players' payoffs reach and remain above the threshold, and then design a state feedback controller such that any initial state reach the set Φ^* in minimum time.

According to the canonical factorization induced by STP, for any integer $1 \leq j \leq k^n$, there exist unique integers $1 \leq \alpha \leq k^v$ and $1 \leq \beta \leq k^{n-v}$ such that $\delta_{k^n}^j = \delta_{k^v}^\alpha \delta_{k^{n-v}}^\beta$. Accordingly, we define the following two sets:

$$\mathcal{Z} = \{\delta_{k^{n-v}}^\beta \mid \delta_{k^n}^j = \delta_{k^v}^\alpha \delta_{k^{n-v}}^\beta, \delta_{k^n}^j \in \Omega\}, \quad (6.13)$$

and

$$\mathcal{U} = \{\delta_{k^v}^\alpha \mid \delta_{k^n}^j = \delta_{k^v}^\alpha \delta_{k^{n-v}}^\beta, \delta_{k^n}^j \in \Omega\}. \quad (6.14)$$

It follows from Definition 6.6 that the control invariant subset Φ is a subset of $\mathcal{Z}$ and that admissible controls are selected only from the set $\mathcal{U}$.

We construct a restricted transition matrix $\hat{L}$ as follows:

$$(\hat{L})_{\beta,j} = \begin{cases} (L)_{\beta,j}, & \text{if } \delta_{k^n}^j \in \Omega \text{ and } \delta_{k^{n-v}}^\beta \in \mathcal{Z}, \\ 0, & \text{otherwise.} \end{cases} \quad (6.15)$$

From the matrix $\hat{L}$, it follows that when $u(t)z(t) \in \Omega$ and $z(t+1) = Lu(t)z(t) = \hat{L}u(t)z(t) \neq \mathbf{0}_{k^{n-v}}$, then $z(t+1) \in \mathcal{Z}$. If $z(t) = \mathbf{0}_{k^{n-v}}$, then $z(t+1) = \hat{L}u(t)z(t) = \hat{L}\mathbf{0}_{k^n} = \mathbf{0}_{k^{n-v}}$ holds for any $u(t) \in \Delta_{k^v}$.

Let $\nu := |\mathcal{U}|$ and $\mathcal{U} := \{\delta_{k^v}^{\gamma_1}, \delta_{k^v}^{\gamma_2}, \dots, \delta_{k^v}^{\gamma_\nu}\}$. We establish a bijective correspondence $\delta_{k^v}^{\gamma_i} \sim \delta_\nu^i$, $i = 1, \dots, \nu$, and denote the reduced control variable by $\bar{u}(t)$. This one-to-one correspondence is described by a function $p : \Delta_{k^v} \rightarrow \Delta_\nu \cup \{\mathbf{0}_\nu\}$ and its algebraic form can be written as

$$\bar{u}(t) = Pu(t), \quad (6.16)$$

where the matrix $P \in \mathbb{R}_{\nu \times k^v}$ is constructed as

$$\text{Col}_\alpha(P) = \begin{cases} \delta_\nu^i, & \text{if } \alpha = \gamma_i, \ i = 1, \dots, \nu, \\ \mathbf{0}_\nu, & \text{otherwise.} \end{cases}$$

Divide the matrix $\hat{L}$ equally into k^v blocks and we select the γ_i -th block, where $i = 1, 2, \dots, \nu$, to yield the following matrix,

$$\tilde{L} = [\text{Blk}_{\gamma_1}(\hat{L}), \text{Blk}_{\gamma_2}(\hat{L}), \dots, \text{Blk}_{\gamma_\nu}(\hat{L})]. \quad (6.17)$$

That is, $\text{Blk}_i(\tilde{L}) := \text{Blk}_{\gamma_i}(\hat{L})$. Construct the matrix

$$\hat{F} := \sum_{l=1}^{\nu} \text{Blk}_l(\tilde{L}). \quad (6.18)$$

It follows that $\hat{F}$ collects all one-step transition information of the states in $\mathcal{Z}$ under admissible controls from the set $\mathcal{U}$. Then, we have the following result to verify whether a given state set is a control invariant subset for the controlled heterogeneous HNEG.

Theorem 6.8. *For system (6.10), the set Φ is a control invariant subset generated*

by Ω if and only if for any $\delta_{k^{n-v}}^\beta \in \Phi$, it holds that

$$\text{Col}_\beta(\widehat{F}^{(\tau)}) \succ \mathbf{0}_{k^{n-v}}, \quad (6.19)$$

where $\tau := |\mathcal{Z}|$.

Proof. Definition 6.6 implies that state $z(0) = \delta_{k^{n-v}}^\beta \in \Phi$ if and only if suitable controls $u(t)$ can be found such that $u(t)z(t) \in \Omega$ for any integer $t \geq 0$. Therefore, it suffices to show that the trajectory of $u(t)z(t)$ will always remain in the set Ω starting from state $z(0) = \delta_{k^{n-v}}^\beta \in \Phi$, if and only if condition (6.19) holds.

(Necessity) Without loss of generality, for a state $z(0) = \delta_{k^{n-v}}^{\beta_0} \in \Phi$, there exists a control sequence $\{u(0), u(1), \dots, u(\tau-1), \dots\} = \delta_{k^v}\{\alpha_0, \alpha_1, \dots, \alpha_{\tau-1}, \dots\}$ to guarantee the trajectory of $u(t)z(t) \in \Omega$ with integer $t \geq 0$ will remain in Ω . Since there are exactly τ states in the set $\mathcal{Z}$, we focus on the system evolution within τ steps:

$$z(0) = \delta_{k^{n-v}}^{\beta_0}, \quad z(1) = \delta_{k^{n-v}}^{\beta_1} = L\delta_{k^v}^{\alpha_0}\delta_{k^{n-v}}^{\beta_0}, \quad \dots, \quad z(\tau) = \delta_{k^{n-v}}^{\beta_\tau} = L\delta_{k^v}^{\alpha_{\tau-1}}\delta_{k^{n-v}}^{\beta_{\tau-1}}.$$

Clearly, $\delta_{k^v}^{\alpha_i}\delta_{k^{n-v}}^{\beta_i}$ belongs to the set Ω for $i = 1, 2, \dots, \tau-1$ and $\delta_{k^v}^{\alpha_i} \in \mathcal{U}$. Define $\delta_{k^v}^{\alpha_i} := \delta_{k^v}^{\lambda_{j_i}^i}$. Hence,

$$\delta_{k^{n-v}}^{\beta_{i+1}} = L\delta_{k^v}^{\alpha_i}\delta_{k^{n-v}}^{\beta_i} = \widehat{L}\delta_{k^v}^{\alpha_i}\delta_{k^{n-v}}^{\beta_i} = \widehat{L}\delta_{k^v}^{\lambda_{j_i}^i}\delta_{k^{n-v}}^{\beta_i} = \widetilde{L}\delta_{k^v}^{j_i}\delta_{k^{n-v}}^{\beta_i}. \quad (6.20)$$

Multiplying both sides of (6.20) by $(\delta_{k^{n-v}}^{\beta_{i+1}})^\top$ yields

$$(\widetilde{L}\delta_{k^v}^{j_i})_{\beta_{i+1}, \beta_i} = 1, \quad \forall i = 0, 1, \dots, \tau-1.$$

Since $\widehat{F} = \widetilde{L} \times \mathbf{1}_\nu = \widetilde{L}(\delta_\nu^1 + \delta_\nu^2 + \dots + \delta_\nu^\nu)$, which implies that

$$(\widehat{F})_{\beta_{i+1}, \beta_i} \geq (\widetilde{L}\delta_{k^v}^{j_i})_{\beta_{i+1}, \beta_i},$$

then we have $(\widehat{F})_{\beta_1, \beta_0} \geq 1$, $(\widehat{F})_{\beta_2, \beta_1} \geq 1$, $\dots$, $(\widehat{F})_{\beta_\tau, \beta_{\tau-1}} \geq 1$, further,

$$(\widehat{F}^{(l)})_{\beta_l, \beta_0} = \sum_{i=1}^{k^{n-v}} (\widehat{F})_{\beta_l, i} (\widehat{F}^{(l-1)})_{i, \beta_0} \geq (\widehat{F})_{\beta_l, \beta_{l-1}} (\widehat{F}^{(l-1)})_{\beta_{l-1}, \beta_0} \geq 1, \quad l = 2, 3, \dots, \tau.$$

It follows that $\text{Col}_{\beta_0}(\widehat{F}^{(\tau)}) \succ \mathbf{0}_{k^{n-v}}$. Therefore, for any $\delta_{k^{n-v}}^\beta \in \Phi$, $\text{Col}_\beta(\widehat{F}^{(\tau)}) \succ \mathbf{0}_{k^{n-v}}$.

(Sufficiency) According to the construction of matrix $\widehat{F}$, we know that each entry of $\widehat{F}$ is either a positive integer or 0. If (6.19) holds, there must exist an index

$l \in \{1, 2, \dots, k^{n-v}\}$ such that $(\hat{F}^\tau)_{l,\beta} \geq 1$. Further, there exist positive integers $\beta_1, \beta_2, \dots, \beta_{\tau-1}$ such that

$$(\hat{F})_{\beta_1,\beta} \geq 1, (\hat{F})_{\beta_2,\beta_1} \geq 1, \dots, (\hat{F})_{l,\beta_{\tau-1}} \geq 1.$$

According to the inequalities above, columns $\bar{u}(0) = \delta_\nu^{j_0}$, $\bar{u}(1) = \delta_\nu^{j_1}$, $\dots$, $\bar{u}(\tau-1) = \delta_\nu^{j_{\tau-1}}$ can be found such that

$$\delta_{k^{n-v}}^{\beta_1} = \tilde{L}\delta_\nu^{j_0}\delta_{k^{n-v}}^\beta \neq \mathbf{0}_{k^{n-v}}, \dots, \delta_{k^{n-v}}^l = \tilde{L}\delta_\nu^{j_{\tau-1}}\delta_{k^{n-v}}^{\beta_{\tau-1}} \neq \mathbf{0}_{k^{n-v}}.$$

From (6.16), we have $u(t) = P^T P u(t) = P^T \bar{u}(t)$. Hence, there exist $u(0), u(1), \dots, u(\tau-1)$ such that

$$\begin{aligned} \delta_{k^{n-v}}^{\beta_1} &= \tilde{L}\delta_\nu^{j_0}\delta_{k^{n-v}}^\beta = \tilde{L}P u(0)\delta_{k^{n-v}}^\beta = \hat{L}u(0)\delta_{k^{n-v}}^\beta, \\ &\vdots \\ \delta_{k^{n-v}}^l &= \tilde{L}\delta_\nu^{j_{\tau-1}}\delta_{k^{n-v}}^{\beta_{\tau-1}} = \tilde{L}P u(\tau-1)\delta_{k^{n-v}}^{\beta_{\tau-1}} = \hat{L}u(\tau-1)\delta_{k^{n-v}}^{\beta_{\tau-1}}, \end{aligned}$$

where $\delta_{k^{n-v}}^{\beta_i} \neq \mathbf{0}_{k^{n-v}}$ with $i = 1, 2, \dots, \tau-1$. There are $\tau+1$ states that are not equal to $\mathbf{0}_{k^{n-v}}$, all of which belong to the set $\mathcal{Z}$. However, the set includes τ states. Thus, at least two states are equal among $\delta_{k^{n-v}}^\beta, \delta_{k^{n-v}}^{\beta_1}, \dots, \delta_{k^{n-v}}^{\beta_{\tau-1}}$ and $\delta_{k^{n-v}}^l$, which implies that a cycle is formed during the system evolution starting from state $\delta_{k^{n-v}}^\beta$ under a control sequence. Therefore, for the initial state $z(0) = \delta_{k^{n-v}}^\beta \in \Phi$, the trajectory of $u(t)z(t)$ will forever stay in the set Ω . $\square$

Remark 6.9. While [124] investigated coupled evolutionary public goods games under payoff threshold constraints, the present work considers the more generalized HNEGs characterized by degree and order heterogeneity. Furthermore, compared to their results on subset invariance, the proposed approach significantly reduces the computational burden in two aspects: (i) by focusing on the control subset $\mathcal{U}$, the number of summation terms required for the construction of matrix $\hat{F}$ is drastically diminished; (ii) our necessary and sufficient condition only requires powers of $\hat{F}$ up to the cardinality of the state subset $\mathcal{Z}$, rather than the entire state space, thereby enhancing the efficiency of the invariance analysis.

Based on Theorem 6.8 and its proof, the maximum control invariant subset Φ^* can be computed via Algorithm 9.

We then examine whether any state in $\Delta_{k^{n-v}} \setminus \Phi^*$ can be steered into the set Φ^* under suitable control sequences. To this end, the aggregate indicator vector of Φ^*

Algorithm 9 Computation of the Maximum Control Invariant Subset Φ^* **Require:** Matrix L and set Ω **Ensure:** The maximum control invariant subset Φ^*

- 1: Compute the state subset $\mathcal{Z}$ and the control subset $\mathcal{U}$ based on (6.13) and (6.14).
- 2: **Initialization:** Set $\tau \leftarrow |\mathcal{Z}|$, $\nu \leftarrow |\mathcal{U}|$, and $\Phi^* \leftarrow \emptyset$.
- 3: Construct the restricted transition matrices $\hat{L}$, $\tilde{L}$, and $\hat{F}$ according to equations (6.15)–(6.18).
- 4: **for** each state $\delta_{k^{n-v}}^\beta \in \mathcal{Z}$ **do**
- 5: **if** $\text{Col}_\beta(\hat{F}^\tau) \succ \mathbf{0}_{k^{n-v}}$ **then**
- 6: Update the subset: $\Phi^* \leftarrow \Phi^* \cup \{\delta_{k^{n-v}}^\beta\}$.
- 7: **end if**
- 8: **end for**
- 9: **return** Φ^*

is defined as

$$V_{\Phi^*} := \sum_{\delta_{k^{n-v}}^{\beta_i} \in \Phi^*} \delta_{k^{n-v}}^{\beta_i}. \quad (6.21)$$

Let

$$\tilde{M} := \sum_{i=1}^{k^v} \text{Blk}_i(L) \quad \text{and} \quad \tilde{F} := \sum_{l=1}^{k^{n-v}-1} \tilde{M}^{(l)}.$$

The matrix $\tilde{F}$ is the controllability matrix of system (6.10), characterizing all possible state transitions in $\Delta_{k^{n-v}}$ that can be induced by admissible control sequences within at most $k^{n-v} - 1$ steps. This observation leads to the following theorem.

Theorem 6.10. *For any state $\delta_{k^{n-v}}^\beta \in \Delta_{k^{n-v}} \setminus \Phi^*$, there exists a finite control sequence that steers the state into Φ^* if and only if*

$$(V_{\Phi^*})^T \text{Col}_\beta(\tilde{F}) > 0, \quad \forall \delta_{k^{n-v}}^\beta \in \Delta_{k^{n-v}} \setminus \Phi^*. \quad (6.22)$$

Proof. Note that $\tilde{F}$ is a nonnegative matrix encoding one-step transitions under admissible controls. Consequently, for any $l \geq 1$, $(\tilde{F}^l)_{j,i} > 0$ if and only if there exists an admissible control sequence of length l that steers the state from $\delta_{k^{n-v}}^i$ to $\delta_{k^{n-v}}^j$. Hence, with $\tilde{F} = \sum_{l=1}^{k^{n-v}-1} \tilde{M}^{(l)}$, we have $(\tilde{F})_{j,i} > 0$ if and only if the state $\delta_{k^{n-v}}^i$ can reach $\delta_{k^{n-v}}^j$ within at most $k^{n-v} - 1$ steps.

Moreover, since the state space $\Delta_{k^{n-v}}$ is finite, any reachable trajectory can be reduced to a simple path by removing cycles. Thus, the existence of a finite control sequence that reaches a destination state is equivalent to the existence of one with length at most $k^{n-v} - 1$. Now fix an arbitrary $\delta_{k^{n-v}}^\beta \in \Delta_{k^{n-v}} \setminus \Phi^*$. There exists a finite admissible control sequence that drives $\delta_{k^{n-v}}^\beta$ into Φ^* if and only if there exists

some $\delta_{k^{n-v}}^{\beta_i} \in \Phi^*$ such that $\delta_{k^{n-v}}^\beta$ can reach $\delta_{k^{n-v}}^{\beta_i}$, i.e.,

$$(\delta_{k^{n-v}}^{\beta_i})^T \text{Col}_\beta(\tilde{F}) > 0.$$

Using equation (6.21), we obtain

$$(V_{\Phi^*})^T \text{Col}_\beta(\tilde{F}) = \sum_{\delta_{k^{n-v}}^{\beta_i} \in \Phi^*} (\delta_{k^{n-v}}^{\beta_i})^T \text{Col}_\beta(\tilde{F}) \geq (\delta_{k^{n-v}}^{\beta_i})^T \text{Col}_\beta(\tilde{F}) > 0.$$

This completes the proof. $\square$

Theorems 6.8 and 6.10 guarantee the existence of a controller. We now turn to the controller design. Specifically, we consider the state feedback control law of the form:

$$u(t) = Kz(t), \quad (6.23)$$

where $K \in \mathcal{L}_{k^v \times k^{n-v}}$ is the state feedback gain matrix. The controller design consists of two stages and follows the truth matrix approach proposed in [94].

The first stage aims to design the state feedback controller (6.23) to guarantee the invariance of the target set Φ^* . To this end, we construct the following truth matrix:

$$(\hat{T})_{j,\beta} = \begin{cases} 1, & \text{if } \tilde{L}\delta_\nu^j \delta_{k^{n-v}}^\beta \in \Phi^* \text{ and } \delta_{k^{n-v}}^\beta \in \Phi^*, \\ 0, & \text{otherwise.} \end{cases} \quad (6.24)$$

The matrix $\hat{T} \in \mathcal{B}_{\nu \times k^{n-v}}$, whose rows and columns represent the controls in $\mathcal{U}$ and the states in $\Delta_{k^{n-v}}$, respectively, characterizes the relationship between states and admissible controls.

We then present the following theorem to determine the admissible controller choices for states in the set Φ^* under state feedback controller (6.23).

Theorem 6.11. *Consider system (6.10) under the state feedback controller (6.23). For any initial state $\delta_{k^{n-v}}^\beta \in \Phi^*$, the corresponding state trajectory remains in Φ^* for all time if and only if*

$$\text{Col}_\beta(K) \leq \text{Col}_\beta(P^T \hat{T}), \quad (6.25)$$

where $\text{Col}_\beta(K) \in \Delta_{k^v}$, and the matrix P is defined by equation (6.16).

Proof. Any initial state $\delta_{k^{n-v}}^\beta \in \Phi^*$ remains in Φ^* for all time if and only if, for any time instant t such that $z(t) = \delta_{k^{n-v}}^\beta \in \Phi^*$, the subsequent state $z(t+1)$ also belongs to Φ^* under the state feedback controller $u(t) = Kz(t)$.

Under the state feedback law, the control applied at the state $\delta_{k^{n-v}}^\beta$ is uniquely determined as

$$u(t) = K\delta_{k^{n-v}}^\beta = \text{Col}_\beta(K) := \delta_{kv}^\alpha.$$

Thus, the state $\delta_{k^{n-v}}^\beta$ is invariant if and only if

$$\delta_{kv}^\alpha \delta_{k^{n-v}}^\beta \in \Omega \quad \text{and} \quad z(t+1) = L\delta_{kv}^\alpha \delta_{k^{n-v}}^\beta = \hat{L}\delta_{kv}^\alpha \delta_{k^{n-v}}^\beta \in \Phi^*.$$

Since $\delta_{kv}^\alpha \in \mathcal{U}$, there exists a unique index $j \in \{1, 2, \dots, \nu\}$ such that

$$\delta_\nu^j = P\delta_{kv}^\alpha, \quad \text{i.e.,} \quad \delta_{kv}^\alpha = P^T\delta_\nu^j.$$

Consequently,

$$\hat{L}\delta_{kv}^\alpha \delta_{k^{n-v}}^\beta = \hat{L}P^T\delta_\nu^j \delta_{k^{n-v}}^\beta = \tilde{L}\delta_\nu^j \delta_{k^{n-v}}^\beta \in \Phi^*,$$

which implies $(\hat{T})_{j,\beta} = 1$. By examining all $j \in \{1, 2, \dots, \nu\}$ satisfying $(\hat{T})_{j,\beta} = 1$, denoted by $j_1, j_2, \dots, j_s$, we obtain

$$\text{Col}_\beta(\hat{T}) = \delta_\nu^{j_1} + \delta_\nu^{j_2} + \dots + \delta_\nu^{j_s}.$$

Using the relation $\delta_{kv}^\alpha = P^T\delta_\nu^j$, there exist indices $\alpha_1, \alpha_2, \dots, \alpha_s$ such that $\delta_{kv}^{\alpha_l} = P^T\delta_\nu^{j_l}$, $l = 1, 2, \dots, s$. Hence,

$$\delta_{kv}^{\alpha_1} + \delta_{kv}^{\alpha_2} + \dots + \delta_{kv}^{\alpha_s} = P^T\text{Col}_\beta(\hat{T}) = \text{Col}_\beta(P^T\hat{T}).$$

Therefore, for the state $\delta_{k^{n-v}}^\beta \in \Phi^*$, the state feedback controller must satisfy

$$\text{Col}_\beta(K) \leq \text{Col}_\beta(P^T\hat{T}),$$

which guarantees that the state trajectory remains in Φ^* . This completes the proof. $\square$

The second stage aims to steer any state in $\Delta_{k^{n-v}} \setminus \Phi^*$ into the set Φ^* under the state feedback controller (6.23).

Let $R_0(\Phi^*) := \Phi^*$ and $\Lambda_0 := \Delta_{k^{n-v}} \setminus \Phi^*$. Define the truth matrix

$$(\hat{T}_{R_0(\Phi^*)|\Lambda_0})_{\alpha,\beta} = \begin{cases} 1, & \text{if } L\delta_{kv}^\alpha \delta_{k^{n-v}}^\beta \in \Phi^* \text{ and } \delta_{k^{n-v}}^\beta \in \Lambda_0, \\ 0, & \text{otherwise,} \end{cases} \quad (6.26)$$

where $\hat{T}_{R_0(\Phi^*)|\Lambda_0} \in \mathcal{B}_{k^v \times k^{n-v}}$. Based on the above truth matrix, we define the set of

states that can reach Φ^* in exactly one step under suitable controls as

$$R_1(\Phi^*) = \{\delta_{k^{n-v}}^\beta \mid \text{Col}_\beta(\hat{T}_{R_0(\Phi^*)|\Lambda_0}) \succ \mathbf{0}_{k^v}\}.$$

Since all states in $R_1(\Phi^*)$ originate from Λ_0 , the set $R_1(\Phi^*)$ is referred to as *the one-step reachable set* of Φ^* . Similarly, for $t \geq 1$, define

$$\Lambda_t := \Lambda_{t-1} \setminus R_t(\Phi^*),$$

and construct the truth matrix

$$(\hat{T}_{R_t(\Phi^*)|\Lambda_t})_{\alpha,\beta} = \begin{cases} 1, & \text{if } L\delta_{k^{n-v}}^\alpha \delta_{k^{n-v}}^\beta \in R_t(\Phi^*) \text{ and } \delta_{k^{n-v}}^\beta \in \Lambda_t, \\ 0, & \text{otherwise.} \end{cases} \quad (6.27)$$

where $\hat{T}_{R_t(\Phi^*)|\Lambda_t} \in \mathcal{B}_{k^v \times k^{n-v}}$. Note that if $t = 0$ in equation (6.27), then it becomes equation (6.26). Accordingly, *the $(t+1)$ -step reachable set* of Φ^* is defined as

$$R_{t+1}(\Phi^*) = \{\delta_{k^{n-v}}^\beta \mid \text{Col}_\beta(\hat{T}_{R_t(\Phi^*)|\Lambda_t}) \succ \mathbf{0}_{k^v}\}.$$

To facilitate understanding, the process of constructing the reachable sets is illustrated in Figure 6.2

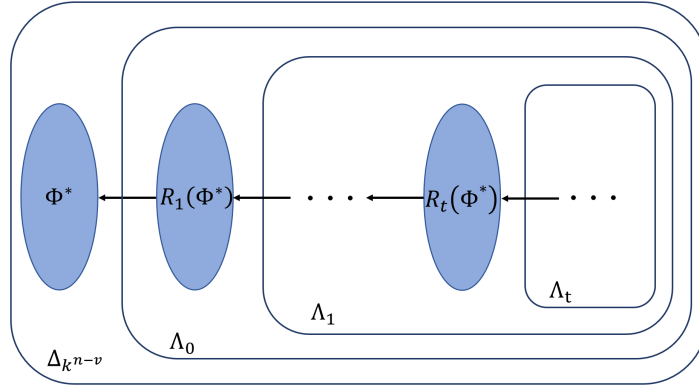


Figure 6.2: The process of finding reachable sets.

Theorem 6.12. Any state $\delta_{k^{n-v}}^\beta \in \Delta_{k^{n-v}} \setminus \Phi^*$ reaches Φ^* in minimum time under the state feedback controller (6.23) if and only if there exists a smallest integer $0 < \xi^* \leq k^{n-v} - |\Phi^*|$ such that

$$\bigcup_{i=1}^{\xi^*} R_i(\Phi^*) = \Delta_{k^{n-v}} \setminus \Phi^*. \quad (6.28)$$

Moreover, for any $\delta_{k^{n-v}}^\beta \in \Delta_{k^{n-v}} \setminus \Phi^*$, the feedback gain matrix K satisfies

$$\text{Col}_\beta(K) \leq \text{Col}_\beta(\tilde{T}), \quad (6.29)$$

where $\text{Col}_\beta(K) \in \Delta_{k^v}$ and $\tilde{T} := \sum_{t=0}^{\xi^*-1} \hat{T}_{R_t(\Phi^*)|\Lambda_t}$.

Proof. (Sufficiency) Suppose that there exists an integer $0 < \xi^* \leq k^{n-v} - |\Phi^*|$ such that equation (6.28) holds and the feedback gain matrix K satisfies equation (6.29). Consider an arbitrary state $\delta_{k^{n-v}}^\beta \in \Delta_{k^{n-v}} \setminus \Phi^*$. By (6.28), there exists an integer $i \in \{1, 2, \dots, \xi^*\}$ such that $\delta_{k^{n-v}}^\beta \in R_i(\Phi^*)$. By the definition of the reachable sets, there exists a control $\delta_{k^v}^\alpha$ such that $L\delta_{k^v}^\alpha \delta_{k^{n-v}}^\beta \in R_{i-1}(\Phi^*)$.

Moreover, since $\delta_{k^{n-v}}^\beta \in R_i(\Phi^*)$, we have $\delta_{k^{n-v}}^\beta \in \Lambda_{i-1}$. By the construction of $\hat{T}_{R_{i-1}(\Phi^*)|\Lambda_{i-1}}$, the nonzero entries of $\text{Col}_\beta(\tilde{T})$ correspond exactly to the admissible controls that drive $\delta_{k^{n-v}}^\beta$ from $R_i(\Phi^*)$ to $R_{i-1}(\Phi^*)$. Therefore, condition (6.29) guarantees that $\text{Col}_\beta(K)$ selects one such admissible control. Further, under the feedback law $u(t) = Kz(t)$, the state evolves from $R_i(\Phi^*)$ to $R_{i-1}(\Phi^*)$. Repeating the same process yields $R_i(\Phi^*) \rightarrow R_{i-1}(\Phi^*) \rightarrow \dots \rightarrow R_0(\Phi^*) := \Phi^*$. Hence, every state in $R_i(\Phi^*)$ reaches Φ^* in i steps.

Moreover, $R_i(\Phi^*) \cap R_j(\Phi^*) = \emptyset$ for any $j \neq i$. Therefore, no state in $R_i(\Phi^*)$ can reach Φ^* in fewer than i steps. It follows that the state reaches Φ^* in minimum time.

Once the state enters Φ^* , Theorem 6.11 guarantees that it remains in Φ^* for all subsequent times. Therefore, every state in $\Delta_{k^{n-v}} \setminus \Phi^*$ reaches Φ^* in minimum time under the state feedback controller (6.23).

(Necessity) Assume that every state $\delta_{k^{n-v}}^\beta \in \Delta_{k^{n-v}} \setminus \Phi^*$ reaches Φ^* in minimum time under the feedback law $u(t) = Kz(t)$. For each state $\delta_{k^{n-v}}^\beta \in \Delta_{k^{n-v}} \setminus \Phi^*$, let i_β denote its minimum reaching time to Φ^* . By the definition of the reachable sets, $\delta_{k^{n-v}}^\beta \in R_{i_\beta}(\Phi^*)$. Since the state space is finite, there exists

$$\xi^* := \max_{\delta_{k^{n-v}}^\beta \in \Delta_{k^{n-v}} \setminus \Phi^*} i_\beta \leq k^{n-v} - |\Phi^*|.$$

Therefore, $\Delta_{k^{n-v}} \setminus \Phi^* = \bigcup_{i=1}^{\xi^*} R_i(\Phi^*)$.

Next, consider an arbitrary state $\delta_{k^{n-v}}^\beta \in R_i(\Phi^*)$. Since the controller drives the state to Φ^* in minimum time, we know from the definition of $R_i(\Phi^*)$ that $L\text{Col}_\beta(K)\delta_{k^{n-v}}^\beta \in R_{i-1}(\Phi^*)$. By the definition of $\hat{T}_{R_{i-1}(\Phi^*)|\Lambda_{i-1}}$, this implies $\text{Col}_\beta(K) \leq \text{Col}_\beta(\hat{T}_{R_{i-1}(\Phi^*)|\Lambda_{i-1}})$. Since $\hat{T}_{R_{i-1}(\Phi^*)|\Lambda_{i-1}} \leq \tilde{T}$, it follows that $\text{Col}_\beta(K) \leq \text{Col}_\beta(\tilde{T})$. $\square$

Finally, according to the proof procedures of Theorems 6.11 and 6.12, the state

feedback gain matrix K is designed via Algorithm 10.

Algorithm 10 Construction of the State Feedback Gain Matrix K

Require: Matrices $L, \tilde{L}$ from (6.17), set Φ^* , and matrix P from (6.16)

Ensure: State feedback gain matrix K

```

1: Initialization: Set  $R_0(\Phi^*) \leftarrow \Phi^*$ ,  $\Lambda_0 \leftarrow \Delta_{k^{n-v}} \setminus \Phi^*$ , and  $t \leftarrow 0$ .
2: if  $\Phi^* = \emptyset$  then
3:   return “No state feedback controller exists.”
4: end if
5: Compute truth matrix  $\hat{T}$  according to (6.24).
6: for each  $\delta_{k^{n-v}}^\beta \in \Phi^*$  do
7:   Select  $\text{Col}_\beta(K) \in \Delta_{k^v}$  such that  $\text{Col}_\beta(K) \leq \text{Col}_\beta(P^T \hat{T})$ .
8: end for
9: while  $\Lambda_t \neq \emptyset$  do
10:  Compute  $\hat{T}_{R_{t-1}(\Phi^*)|\Lambda_{t-1}}$  according to (6.27).
11:   $R_t(\Phi^*) \leftarrow \{\delta_{k^{n-v}}^\beta \mid \text{Col}_\beta(\hat{T}_{R_{t-1}(\Phi^*)|\Lambda_{t-1}}) \succ \mathbf{0}_{k^v}\}$ .
12:  if  $R_{t+1}(\Phi^*) = \emptyset$  then
13:    return “No state feedback controller exists.”
14:  end if
15:   $\Lambda_{t+1} \leftarrow \Lambda_t \setminus R_{t+1}(\Phi^*)$ .
16:   $t \leftarrow t + 1$ .
17: end while
18: Set  $\xi^* \leftarrow t$  and compute  $\tilde{T} \leftarrow \sum_{i=0}^{\xi^*-1} \hat{T}_{R_i(\Phi^*)|\Lambda_i}$ .
19: for each  $\delta_{k^{n-v}}^\beta \in \Delta_{k^{n-v}} \setminus \Phi^*$  do
20:   Select  $\text{Col}_\beta(K) \in \Delta_{k^v}$  such that  $\text{Col}_\beta(K) \leq \text{Col}_\beta(\tilde{T})$ .
21: end for
22: return  $K$ 

```

Remark 6.13. Algorithm 10 provides a constructive and finite-step solution to the strategy optimization problem formulated earlier. Specifically, it constructs the state feedback gain matrix K by iteratively computing the truth matrices $\hat{T}_{R_t(\Phi^*)|\Lambda_t}$ and the reachable sets $R_{t+1}(\Phi^*)$. At each iteration, the truth matrix is obtained by checking all state-control pairs $(\delta_{k^{n-v}}^\beta, \delta_{k^v}^\alpha)$, which incurs a computational complexity of $\mathcal{O}(k^{n-v} \cdot k^v)$. The number of iterations of Algorithm 10 is upper bounded by $k^{n-v} - |\Phi^*|$, since each iteration strictly reduces $|\Lambda_t|$. Therefore, the overall worst-case computational complexity is bounded by $\mathcal{O}(k^{2n-v})$ assuming $|\Delta_{k^v}| = k^v$ is small relative to $|\Delta_{k^{n-v}}| = k^{n-v}$.

6.4 An Example

Example 6.14. Consider a stag hunt game in which hunters coordinate to capture a deer on a heterogeneous hypergraph consisting of six nodes (hunters), labeled by $N = \{1, 2, \dots, 6\}$, and six hyperedges, labeled by $E = \{E^1, E^2, \dots, E^6\}$. Among these hyperedges, three are of order two and three are of order three, as illustrated in Figure 6.3. Each hunter (player) has two available strategies: coordinate (C) and defect (D). The payoffs for mutual cooperation are set to $r_1 = 4$ and $r_2 = 6$, while the payoffs for mutual defection are $p_1 = 3$ and $p_2 = 4$, for the 2-player FNG G_1 and the 3-player FNG G_2 respectively. In the game G_1 , if one hunter cooperates while the other yields to temptation and defects by choosing to hunt a hare, the defector obtains a lower payoff ($t < r_1$), whereas the cooperator hunts in vain and receives a zero payoff ($s = 0 < p_1$). In the game G_2 , two additional payoff parameters are introduced: the payoff for defecting against one cooperator and one defector ($w = 0$), and the payoff for cooperating against one cooperator and one defector ($e = 3$) [130].

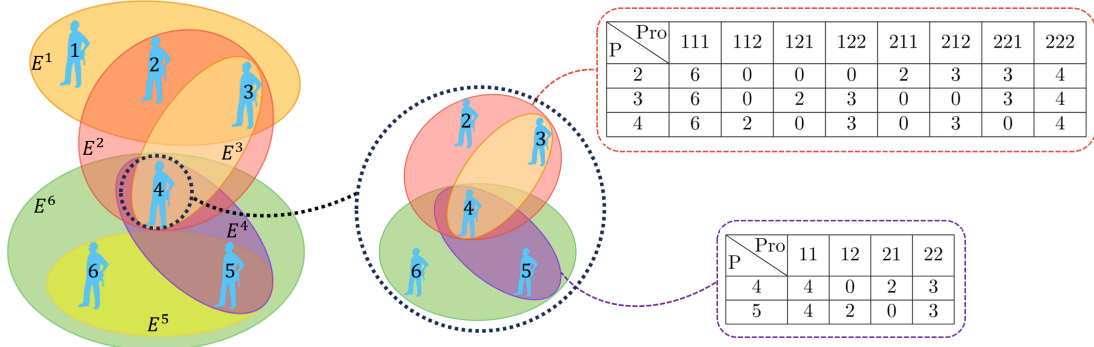


Figure 6.3: Stag hunt game on a hypergraph consisting of six nodes and six hyperedges. The central illustration highlights the local interaction structure of player 4. The overall payoff of the player 4 is an aggregate of payoffs derived from these distinct hyperedges, as detailed in the corresponding payoff matrices for the 2-player and the 3-player FNGs.

Let $C = 1$ and $D = 2$. The payoff matrix of each player on the 2-player FNG G_1 and that on the 3-player FNG G_2 is shown in Figure 6.3.

Now suppose a forest ranger can instruct the first two hunters in their daily hunting strategies, while the remaining four hunters choose strategies myopically based on their own interests. The forest ranger aims to achieve the following goal: the payoff of every hunter reaches a prescribed subsistence level (payoff threshold) $\tilde{P} = 4$ in minimum time, and once this level is reached, the entire group can maintain a stable status in the long term. Thus, we assume that players 1 and 2 are control

players and then design the strategies for control players such that the payoff of each player reaches and remains above $\tilde{P} = 4$ in minimum time.

According to the payoff matrix shown in Figure 6.3, we obtain the payoff vectors of player i on hyperedges of order g . Specifically, for the game G_1 , $V_1^2 = [4, 0, 2, 3]$, $V_2^2 = [4, 2, 0, 3]$, and for the game G_2 , $V_1^3 = [6, 0, 0, 0, 2, 3, 3, 4]$, $V_2^3 = [6, 0, 2, 3, 0, 0, 3, 4]$, and $V_3^3 = [6, 2, 0, 3, 0, 3, 0, 4]$. Furthermore, according to Eqs. (6.2), (6.3) and (6.7), we derive the algebraic form of payoff for player 1 as follows:

$$c_1(t) = V_1^3 x_1(t) x_2(t) x_3(t) = V_1^3 D_f^{[8,8]} x(t) := \overline{W}_1 x_1(t) x_{-1}(t),$$

where

$$\begin{aligned} \overline{W}_1 = & [6, 6, 6, 6, 6, 6, 6, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \\ & 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, \\ & 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4]. \end{aligned}$$

Similarly, the structure vectors of the payoff for the remaining players can be calculated and are omitted here due to space limitations.

According to myopic best response adjustment rule Eqs. (6.5) and (6.8), we can construct matrices L_i with $i = 3, 4, 5, 6$. Due to space limitations, the explicit expressions are omitted.

We redefine the strategies of the control players (x_1, x_2) as u_1 and u_2 , and the strategies of the remaining players (x_3, x_4, x_5, x_6) as z_1, z_2, z_3 and z_4 , further, we obtain the algebraic form of the controlled heterogeneous HNEG as

$$z(t+1) = Lu(t)z(t), \quad (6.30)$$

where

$$\begin{aligned} L = \delta_{16} [& 1, 3, 2, 8, 12, 12, 12, 16, 1, 7, 6, 8, 12, 16, 16, 16, 9, \\ & 15, 14, 16, 12, 16, 16, 16, 13, 15, 14, 16, 16, 16, 16, 16, \\ & 1, 3, 2, 8, 12, 12, 12, 16, 9, 15, 14, 16, 12, 16, 16, 9, \\ & 15, 14, 16, 12, 16, 16, 16, 13, 15, 14, 16, 16, 16, 16, 16]. \end{aligned}$$

According to Eqs. (6.3), (6.11) and (6.12), we compute the strategy profile set under which the payoff of each player exceeds the prescribed threshold $\tilde{P} = 4$ as follows:

$$\Omega = \{\delta_{64}^1, \delta_{64}^2, \delta_{64}^4, \delta_{64}^5, \delta_{64}^8, \delta_{64}^{57}, \delta_{64}^{58}, \delta_{64}^{61}, \delta_{64}^{64}\}. \quad (6.31)$$

Next, we design a state feedback controller $u(t) = Hy(t)$ such that any initial

profile $u(t)z(t)$ converges to the set Ω in minimum time T . We first calculate the maximum control invariant subset generated by Ω .

The set Ω can be divided into $\mathcal{Z} = \{\delta_{16}^1, \delta_{16}^2, \delta_{16}^4, \delta_{16}^5, \delta_{16}^8, \delta_{16}^9, \delta_{16}^{10}, \delta_{16}^{13}, \delta_{16}^{16}\}$ and $\mathcal{U} = \{\delta_4^1, \delta_4^4\}$. Then, we know from Algorithm 9 that the maximum control invariant subset is $\Phi^* = \{\delta_{16}^1, \delta_{16}^4, \delta_{16}^8, \delta_{16}^9, \delta_{16}^{13}, \delta_{16}^{16}\}$.

By constructing the truth matrix

$$(\hat{T})_{j,\beta} = \begin{cases} 1, & \text{if } \tilde{L}\delta_{16}^j\delta_{16}^\beta \in \Phi^*, \delta_{16}^\beta \in \Phi^*, \\ 0, & \text{otherwise,} \end{cases}$$

and calculate the matrix

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

we can determine several columns of the state feedback gain matrix K , namely,

$$\text{Col}_1(K) = \delta_4^1, \text{Col}_4(K) = \delta_4^1, \text{Col}_8(K) = \delta_4^1,$$

$$\text{Col}_9(K) = \delta_4^4, \text{Col}_{13}(K) = \delta_4^4, \text{Col}_{16}(K) = \delta_4^4.$$

Moreover, let $R_0(\Phi^*) := \Phi^*$ and $\Lambda_0 := \Delta_{16} \setminus \Phi^*$. By constructing the truth matrix

$$(\hat{T}_{R_0(\Phi^*)|\Lambda_0})_{\alpha,\beta} = \begin{cases} 1, & \text{if } L\delta_4^\alpha\delta_{16}^\beta \in \Phi^*, \delta_{16}^\beta \in \Lambda_0, \\ 0, & \text{otherwise,} \end{cases}$$

we have the one-step reachable set of Φ^* , i.e., $R_1(\Phi^*) = \{\delta_{16}^6, \delta_{16}^7, \delta_{16}^{12}, \delta_{16}^{14}, \delta_{16}^{15}\}$. And $\Lambda_1 := \Lambda_0 \setminus R_1(\Phi^*) = \{\delta_{16}^2, \delta_{16}^3, \delta_{16}^5, \delta_{16}^{10}, \delta_{16}^{11}\}$. Then, construct the truth matrix

$$(\hat{T}_{R_1(\Phi^*)|\Lambda_1})_{\alpha,\beta} = \begin{cases} 1, & \text{if } L\delta_4^\alpha\delta_{16}^\beta \in R_1(\Phi^*), \delta_{16}^\beta \in \Lambda_1, \\ 0, & \text{otherwise.} \end{cases}$$

Accordingly, $R_2(\Phi^*) = \{\delta_{16}^2, \delta_{16}^3, \delta_{16}^5, \delta_{16}^{10}, \delta_{16}^{11}\}$. It follows that

$$R_1(\Phi^*) + R_2(\Phi^*) = \Delta_{16} \setminus \Phi^*,$$

and hence the minimum convergence time is $\xi^* = 2$. The remaining columns of the matrix K can be designed according to Algorithm 10 as follows:

$$\text{Col}_2(K) \in \{\delta_4^2, \delta_4^4\}, \text{Col}_3(K) \in \{\delta_4^2, \delta_4^4\}, \text{Col}_5(K) \in \{\delta_4^1, \delta_4^2, \delta_4^3, \delta_4^4\},$$

$$\text{Col}_6(K) \in \{\delta_4^2, \delta_4^4\}, \dots, \text{Col}_{15}(K) \in \{\delta_4^1, \delta_4^2, \delta_4^3, \delta_4^4\}.$$

Therefore, the state feedback gain matrix K can be designed as

$$K = \delta_4[1, i_2, i_3, 1, i_5, i_6, i_7, 1, 4, i_{10}, i_{11}, i_{12}, 4, i_{14}, i_{15}, 4] \quad (6.32)$$

where $i_2, i_3, i_6, i_7, i_{12} \in \{2, 4\}$ and $i_5, i_{10}, i_{11}, i_{14}, i_{15} \in \{1, 2, 3, 4\}$, such that the cost function satisfies $J(u, z_0) = 2$. This implies that the payoff of each player exceeds the threshold $\tilde{P} = 4$ in minimum time.

Specifically, we select one state feedback gain matrix $K = \delta_4[1, 2, 2, 1, 1, 2, 2, 1, 4, 4, 1, 2, 4, 3, 3, 4]$ from equation (6.32). Under this controller, the state evolution trajectories of system (6.30) starting from any initial state are illustrated in Figure 6.4. The corresponding strategy evolution of the six players is shown in Figure 6.5, where the blue hollow circles and solid circles denote strategies 1 and 2 respectively. Here, we only present the cases with initial states $\delta_{16}^1, \delta_{16}^2, \delta_{16}^3$, and δ_{16}^4 due to space limitations.

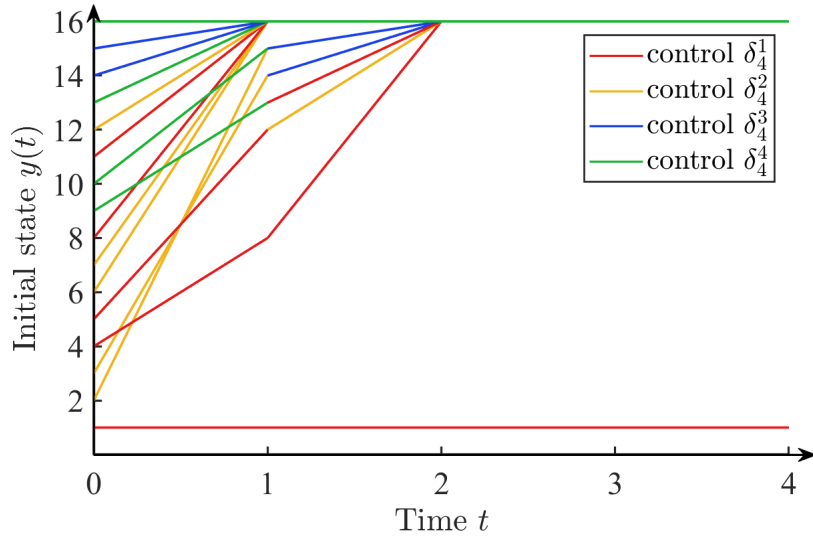


Figure 6.4: State evolution trajectory from every initial state under the state feedback controller $u(t) = \delta_4[1, 2, 2, 1, 1, 2, 2, 1, 4, 4, 1, 2, 4, 3, 3, 4]z(t)$, where i on the z -axis corresponds to $z(t) = \delta_{16}^i$.

6.5 Conclusion

This chapter investigated the strategy optimization problem for a class of controlled heterogeneous HNEG with structural heterogeneities. First, the algebraic

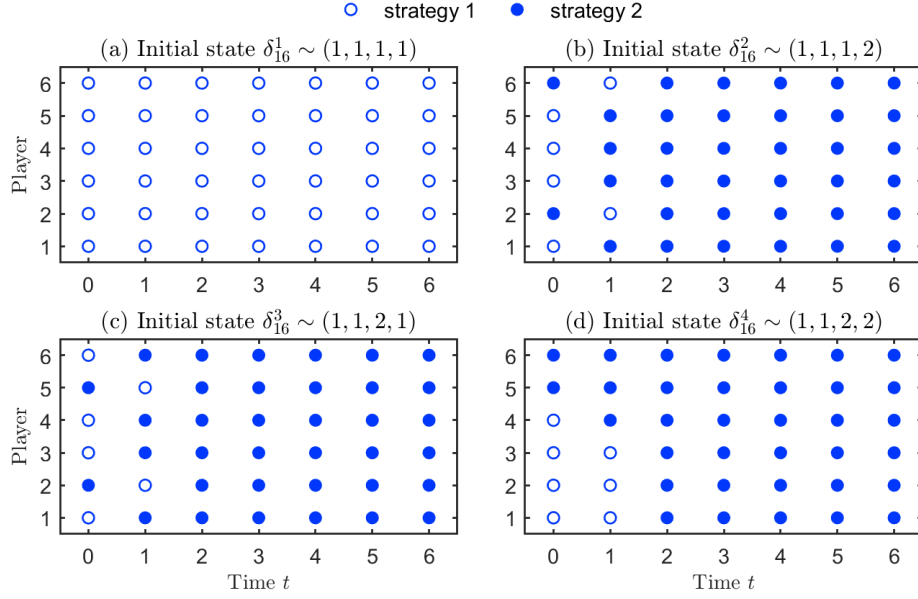


Figure 6.5: Strategy evolution of each player under the state feedback controller $u(t) = \delta_4[1, 2, 2, 1, 1, 2, 2, 1, 4, 4, 1, 2, 4, 3, 3, 4]z(t)$.

representation of the payoff functions was derived, based on which an algebraic framework of heterogeneous HNEGs was established using the STP approach. By introducing control players into the game, the resulting model can be viewed as a controlled heterogeneous HNEG and the optimization problem was converted into an equivalent minimum-time set controllability problem. Necessary and sufficient conditions for set controllability and control invariance were provided. Moreover, a state feedback controller was designed based on truth matrices, guaranteeing that all players' payoff reach and remain above the prescribed threshold in minimum time. A stag hunt game on a heterogeneous hypergraph example validated the feasibility of the proposed control design.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis has investigated the identification and control of logical dynamical systems within a unified algebraic framework based on the STP. The main focus has been on addressing fundamental problems arising in identification of SBNs, ROTP of BNs and SBNs, and strategy optimization of heterogeneous HNEGs.

First, the identification problem for SBNs has been studied under general switching structures. By establishing a unified algebraic representation that captures the joint effects of state transitions, outputs, and switching signals, conditions for system identifiability have been derived. The results reveal that the identifiability of the overall system depends not only on the properties of individual subnetworks but also on the interaction between switching mechanisms and observation structures.

Second, the notion of the ROT has been introduced for BNs and BCNs. This formulation relaxes the conventional requirements of exact and periodic tracking and provides a more flexible framework for describing long-term system behaviors. For BN and BCNs, algebraic conditions and constructive controller design methods have been developed to ensure recurrent tracking performance.

Third, the ROTP has been further extended to SBCNs. By reformulating the problem in terms of set controllability, a systematic approach has been established to jointly design control inputs and switching signals. The results highlight the critical role of switching mechanisms in shaping system reachability and tracking capabilities.

Finally, the proposed algebraic framework has been applied to evolutionary game systems with higher-order interactions. By modeling HNEGs as logical dynamical systems, the strategy optimization problem has been transformed into a control

problem under constraints. This demonstrates the effectiveness and generality of the developed framework in addressing complex multi-player interaction systems beyond traditional BNs.

Overall, this thesis provides a unified perspective for modeling, analysis, and control of logical dynamical systems, bridging BNs, SBNs, and HNEGs within a common algebraic framework.

7.2 Future Work

Although significant progress has been made in this thesis, several directions remain open for future investigation.

- 1). Identification under partial observation and noisy data: The identification results developed in this thesis are mainly based on ideal observation settings. In practical applications, measurements are often incomplete or corrupted by noise. Extending the identification framework to scenarios with partial observability, output uncertainty, or stochastic disturbances is an important and challenging direction.
- 2). Scalability and computational complexity: As the network size increases, the state space of BNs grows exponentially, leading to significant computational challenges. Developing scalable algorithms and efficient computational techniques for large-scale systems is essential for practical applications.
- 3). Extension to more complex ROTP and general logical systems: Generalizing the results of ROT to cases where the reference signal evolves dynamically is an important direction for future research. Such extensions introduce additional complexities, as the current methodologies for designing state feedback controllers are not directly applicable. New approaches will be required to address the dynamic evolution of the reference signal and analyze the associated recurrent behaviors. Moreover, studying ROTP for specialized network types, such as those with restricted states and controls or faulty structures, could also be a meaningful research direction, which promises to significantly expand the applicability of ROTP to more complex and realistic systems.
- 4). Output feedback and dynamic control design: The control strategies considered in this thesis are primarily based on state feedback. However, in many real-world systems, full state information may not be available. The development

of output feedback control and other convenient methods for ROTP of BNs, particularly under switching environments, remains an open problem.

- 5). Advanced strategy optimization in HNEGs: For HNEGs, further research can be conducted on more general strategy update rules, distributed optimization methods, and learning-based approaches. In particular, incorporating uncertainty, time delays, and adaptive behaviors into the control and optimization framework would enhance its applicability to real-world systems. Additionally, the number of control players is predetermined in this thesis, thus finding the minimum number of control players to optimize the game is a topic worthy of further research.

Appendix A

List of Publications

A.1 Author's Publication

1. **C. Jiang**, B. Wang, C. Del Vecchio, & J. Feng (2025). Identification of switched Boolean networks. *Information Sciences*, 719, 122431.
2. **C. Jiang**, S. Fu, C. Del Vecchio, B. Wang, J. Wang, & J. Zhao (2025). Recurrent output tracking of Boolean networks. *Nonlinear Analysis: Hybrid Systems*, 56, 101583.
3. **C. Jiang**, B. Wang, S. Fu, J. Zhao, & M. Sun (2023). Observability of periodically switched Boolean control networks. *Journal of Systems Science and Complexity*, 36(3), 985-1001.
4. Y. Chai, **C. Jiang**, Y. Wu, & C. Del Vecchio (2026). Algebraic expressions for stochastic dynamics in populations: from strategy profile to aggregate state. *Science China: Information Sciences*, 69(4), 140208.
5. S. Fu, Y. Zhao, **C. Jiang**, & J. Xu (2024). Pinning control design for the output regulation of Boolean networks. *International Journal of Control*, 98(7), 1495-1504.
6. **C. Jiang**, S. Fu, Y. Wu, & C. Del Vecchio (2025, July). Recurrent output tracking of switched Boolean control networks under constant reference signal. In *2025 American Control Conference (ACC)* (pp. 1938-1943). IEEE.
7. **C. Jiang**, C. Del Vecchio, & B. Wang (2025, July). Switched Boolean network identification under multiple samples. In *2025 11th International Conference on Control, Decision and Information Technologies (CoDIT)* (pp. 671-676).

-
8. **C. Jiang**, Y. Chai, S. Fu, Y. Wu, & C. Del Vecchio (2026). Strategy optimization of controlled heterogeneous hyper-networked evolutionary games. *IEEE Transactions on Control of Network Systems*. (Under Review)
 9. S. Zou, **C. Jiang**, J. Zhu, & C. Del Vecchio (2026). Synchronization of linear networks with switching topologies over finite fields. *IEEE Transactions on Automatic Control*. (Under Review)
-

Bibliography

- [1] S. Kauffman, “Metabolic stability and epigenesis in randomly constructed genetic nets,” *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.
- [2] T. Ideker, T. Galitski, and L. Hood, “A new approach to decoding life: systems biology,” *Annual Review of Genomics and Human Genetics*, vol. 2, no. 1, pp. 343–372, 2001.
- [3] M. Villani and R. Serra, “On the dynamical properties of a model of cell differentiation,” *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2013, no. 1, p. 4, 2013.
- [4] A. Roli, M. Manfroni, C. Pinciroli, and M. Birattari, “On the design of boolean network robots,” in *European Conference on the Applications of Evolutionary Computation*. Springer, 2011, pp. 43–52.
- [5] J. Alexander, “Random Boolean networks and evolutionary game theory,” *Philosophy of Science*, vol. 70, no. 5, pp. 1289–1304, 2003.
- [6] J. Tyson, A. Csikasz-Nagy, and B. Novak, “The dynamics of cell cycle regulation,” *BioEssays*, vol. 24, no. 12, pp. 1095–1109, 2002.
- [7] H. Li and Y. Wang, “On reachability and controllability of switched Boolean control networks,” *Automatica*, vol. 48, no. 11, pp. 2917–2922, 2012.
- [8] S. Fu, J. Zhao, and J. Wang, “Input-output decoupling control design for switched Boolean control networks,” *Journal of the Franklin Institute*, vol. 355, no. 17, pp. 8576–8596, 2018.
- [9] H. Li, Y. Wang, and Z. Liu, “Stability analysis for switched Boolean networks under arbitrary switching signals,” *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1978–1982, 2014.

- [10] A. Yerudkar, C. Del Vecchio, and L. Glielmo, “Output tracking control design of switched Boolean control networks,” *IEEE Control Systems Letters*, vol. 4, no. 2, pp. 355–360, 2020.
 - [11] Y. Sun and J. Zhao, “Passivity-based stabilization for switched stochastic nonlinear systems,” *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 5, pp. 1307–1309, 2024.
 - [12] K. Zhang, L. Zhang, and L. Xie, “Finite automata approach to observability of switched Boolean control networks,” *Nonlinear Analysis: Hybrid Systems*, vol. 19, pp. 186–197, 2016.
 - [13] J. Smith and G. Price, “The logic of animal conflict,” *Nature*, vol. 246, no. 5427, pp. 15–18, 1973.
 - [14] M. Nowak and R. May, “Evolutionary games and spatial chaos,” *Nature*, vol. 359, no. 6398, pp. 826–829, 1992.
 - [15] J. Yan, “Personal sustained cooperation based on networked evolutionary game theory,” *Scientific Reports*, vol. 13, no. 1, p. 9125, 2023.
 - [16] Y. Jia, Y. Li, and J. Feng, “Networked evolutionary game-based demand response via feedback controls,” *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 2, pp. 1638–1646, 2023.
 - [17] A. Sheng, Q. Su, L. Wang, and J. B. Plotkin, “Strategy evolution on higher-order networks,” *Nature Computational Science*, vol. 4, no. 4, pp. 274–284, 2024.
 - [18] I. Papastaikoudis, J. Newton, J. Watson, and I. Lestas, “A hypergraph evolutionary game theoretic Cournot model for networked markets,” in *International Conference on Game Theory for Networks*. Cham, Switzerland: Springer, 2025, pp. 18–41.
 - [19] Y. Zhu, H. Cui, and C. Xia, “Adaptive reputation promotes the cooperation of multiplayer snowdrift game on higher-order networks,” *Physica A: Statistical Mechanics and its Applications*, p. 130824, 2025.
 - [20] A. Civilini, N. Anbarci, and V. Latora, “Evolutionary game model of group choice dilemmas on hypergraphs,” *Physical Review Letters*, vol. 127, no. 26, p. 268301, 2021.
-

- [21] C. Berge, *Hypergraphs: combinatorics of finite sets*. Elsevier, 1984, vol. 45.
 - [22] J. Wang, L. Rong, and Q. Deng, “Evolving hypernetwork model,” *The European Physical Journal B*, vol. 77, no. 4, pp. 493–498, 2010.
 - [23] E. Estrada and J. Rodriguez-Velazquez, “Subgraph centrality in complex networks,” *Physical Review E*, vol. 71, no. 5, p. 056103, 2005.
 - [24] U. Alvarez-Rodriguez, F. Battiston, G. de Arruda, Y. Moreno, M. Perc, and V. Latora, “Evolutionary dynamics of higher-order interactions in social networks,” *Nature Human Behaviour*, vol. 5, no. 5, pp. 586–595, 2021.
 - [25] B. Lin, L. Zhou, Z. Gao, and H. Fang, “Evolutionary dynamics of group cooperation on heterogeneous higher-order networks,” *IEEE Transactions on Network Science and Engineering*, vol. 12, no. 6, pp. 4894–4905, 2025.
 - [26] M. Aldana, “Boolean dynamics of networks with scale-free topology,” *Physica D: Nonlinear Phenomena*, vol. 185, no. 1, pp. 45–66, 2003.
 - [27] F. Ay, F. Xu, and T. Kahveci, “Scalable steady state analysis of Boolean biological regulatory networks,” *PloS one*, vol. 4, no. 12, p. e7992, 2009.
 - [28] Y. Achdou and I. Capuzzo-Dolcetta, “Mean field games: numerical methods,” *SIAM Journal on Numerical Analysis*, vol. 48, no. 3, pp. 1136–1162, 2009.
 - [29] Y. Yu and Y. Ni, “Network effects of animal conflicts,” *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 25, pp. 6379–6384, 2008.
 - [30] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-tensor Product Approach*. London: Springer, 2011.
 - [31] D. Cheng, “Semi-tensor product of matrices and its application to Morgan’s problem,” *Science in China Series F: Information Sciences*, vol. 44, no. 3, pp. 195–212, 2001.
 - [32] D. Cheng and H. Qi, “A linear representation of dynamics of Boolean networks,” *IEEE Transactions on Automatic Control*, vol. 55, no. 10, pp. 2251–2258, 2010.
 - [33] Z. Liu and H. Li, “Controllability and observability of Boolean control networks subject to stuck-at fault,” *Nonlinear Analysis: Hybrid Systems*, vol. 51, p. 101435, 2024.
-

- [34] D. Cheng and H. Qi, “Controllability and observability of Boolean control networks,” *Automatica*, vol. 45, no. 7, pp. 1659–1667, 2009.
 - [35] Y. Jia, D. Cheng, and J. Feng, “State feedback stabilization of generic logic systems via Ledley antecedence solution,” *Mathematical Methods in the Applied Sciences*, vol. 44, no. 11, pp. 9061–9071, 2021.
 - [36] M. Meng and L. Li, “Stability and pinning stabilization of Markovian jump Boolean networks,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 8, pp. 3565–3569, 2022.
 - [37] B. Chen, X. Yang, Y. Liu, and J. Qiu, “Controllability and stabilization of Boolean control networks by the auxiliary function of flipping,” *International Journal of Robust and Nonlinear Control*, vol. 30, no. 14, pp. 5529–5541, 2020.
 - [38] Y. Li and J. Zhu, “Necessary and sufficient vertex partition conditions for input-output decoupling of Boolean control networks,” *Automatica*, vol. 137, p. 110097, 2022.
 - [39] L. Du, Z. Zhang, and C. Xia, “A node-pinning and state-flipped approach to partial synchronization of Boolean networks,” *Nonlinear Analysis: Hybrid Systems*, vol. 53, p. 101501, 2024.
 - [40] L. Wang and Z. Wu, “Pinning synchronization of large-scale Boolean networks,” *IEEE Transactions on Automatic Control*, vol. 69, no. 5, pp. 3404–3410, 2024.
 - [41] Y. Wang and P. Guo, “Optimal control of singular Boolean control networks via Ledley solution method,” *Journal of the Franklin Institute*, vol. 358, no. 12, pp. 6161–6173, 2021.
 - [42] L. Wu and J. Sun, “Optimal preview pinning control of Boolean networks,” *ISA Transactions*, vol. 146, pp. 291–296, 2024.
 - [43] H. Li, G. Zhao, M. Meng, and J. Feng, “A survey on applications of semi-tensor product method in engineering,” *Science China Information Sciences*, vol. 61, no. 1, pp. 1–17, 2018.
 - [44] G. Beintema, M. Schoukens, and R. Tóth, “Meta-state-space learning: An identification approach for stochastic dynamical systems,” *Automatica*, vol. 167, p. 111787, 2024.
-

- [45] X. Luan, X. Ping, S. Zhao, F. Ding, and F. Liu, “A highly-accurate identification method for linear systems using transferred knowledge,” *Automatica*, vol. 173, p. 112016, 2025.
 - [46] T. Akutsu, S. Miyano, and S. Kuhara, “Identification of genetic networks from a small number of gene expression patterns under the Boolean network model,” in *Pacific Symposium on Biocomputing*, vol. 4, 1999, pp. 17–28.
 - [47] N. Berestovsky and L. Nakhleh, “An evaluation of methods for inferring Boolean networks from time-series data,” *PLoS ONE*, vol. 8, no. 6, p. e66031, 2013.
 - [48] T. Leifeld, Z. Zhang, and P. Zhang, “Identification of Boolean network models from time series data incorporating prior knowledge,” *Frontiers in Physiology*, vol. 9, p. 695, 2018.
 - [49] M. Ostrowski, L. Paulevé, T. Schaub, A. Siegel, and C. Guziolowski, “Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming,” *BioSystems*, vol. 149, pp. 139–153, 2016.
 - [50] D. Cheng, H. Qi, and Z. Li, “Model construction of Boolean network via observed data,” *IEEE Transactions on Neural Networks*, vol. 22, no. 4, pp. 525–536, 2011.
 - [51] D. Cheng and Y. Zhao, “Identification of Boolean control networks,” *Automatica*, vol. 47, no. 4, pp. 702–710, 2011.
 - [52] E. Fornasini and M. Valcher, “Identification problems for Boolean networks and Boolean control networks,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 4544–4549, 2014.
 - [53] X. Zhang, H. Han, and W. Zhang, “Identification of Boolean networks using premined network topology information,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 2, pp. 464–469, 2017.
 - [54] B. Wang, J. Feng, and D. Cheng, “On identification of Boolean control networks,” *SIAM Journal on Control and Optimization*, vol. 60, no. 3, pp. 1591–1612, 2022.
 - [55] T. Mu, J. Feng, B. Wang, and Y. Jia, “Identification of Boolean control networks with time delay,” *ISA Transactions*, vol. 144, pp. 113–123, 2024.
-

- [56] R. Zhao, J. Feng, Q. Meng, and B. Wang, "Identification of a class of singular Boolean control networks," *Information Sciences*, vol. 691, p. 121627, 2025.
 - [57] C. Cobelli and G. Romanin-Jacur, "Controllability, observability and structural identifiability of multi input and multi output biological compartmental systems," *IEEE Transactions on Biomedical Engineering*, no. 2, pp. 93–100, 1976.
 - [58] E. Fornasini and M. Valcher, "Observability, reconstructibility and state observers of Boolean control networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1390–1401, 2013.
 - [59] K. Zhang, "A survey on observability of Boolean control networks," *Control Theory and Technology*, vol. 21, no. 1, pp. 1–14, 2023.
 - [60] Y. Zhao, D. Cheng, and H. Qi, "Input-state incidence matrix of Boolean control networks and its applications," *Systems & Control Letters*, vol. 59, no. 12, pp. 767–774, 2010.
 - [61] D. Laschov, M. Margaliot, and G. Even, "Observability of Boolean networks: A graph-theoretic approach," *Automatica*, vol. 49, no. 8, pp. 2351–2362, 2013.
 - [62] K. Zhang and L. Zhang, "Observability of Boolean control networks: A unified approach based on finite automata," *IEEE Transactions on Automatic Control*, vol. 61, no. 9, pp. 2733–2738, 2016.
 - [63] Y. Yu, M. Meng, and J. Feng, "Observability of Boolean networks via matrix equations," *Automatica*, vol. 111, p. 108622, 2020.
 - [64] Y. Yu, M. Meng, J. Feng, and G. Chen, "Observability criteria for Boolean networks," *IEEE Transactions on Automatic Control*, vol. 67, no. 11, pp. 6248–6254, 2022.
 - [65] D. Cheng, C. Li, and F. He, "Observability of Boolean networks via set controllability approach," *Systems & Control Letters*, vol. 115, pp. 22–28, 2018.
 - [66] Y. Yang, Y. Liu, J. Lou, and Z. Wang, "Observability of switched Boolean control networks using algebraic forms," *Discrete & Continuous Dynamical Systems-Series S*, vol. 14, no. 4, pp. 1519–1533, 2021.
 - [67] L. Zhang, J. Feng, and J. Yao, "Controllability and observability of switched Boolean control networks," *IET Control Theory & Applications*, vol. 6, no. 16, pp. 2477–2484, 2012.
-

- [68] D. Jiang and K. Zhang, “Observability of Boolean control networks with time-variant delays in states,” *Journal of Systems Science and Complexity*, vol. 31, no. 2, pp. 436–445, 2018.
 - [69] F. Li, J. Sun, and Q. Wu, “Observability of Boolean control networks with state time delays,” *IEEE Transactions on Neural Networks*, vol. 22, no. 6, pp. 948–954, 2011.
 - [70] R. Zhou, Y. Guo, and W. Gui, “Set reachability and observability of probabilistic Boolean networks,” *Automatica*, vol. 106, pp. 230–241, 2019.
 - [71] D. Laschov and M. Margaliot, “Controllability of Boolean control networks via the Perron-Frobenius theory,” *Automatica*, vol. 48, no. 6, pp. 1218–1223, 2012.
 - [72] A. Yerudkar, C. Del Vecchio, N. Singh, and L. Glielmo, “Reachability and controllability of delayed switched Boolean control networks,” in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 1863–1868.
 - [73] D. Cheng, C. Li, and F. He, “Controllability of Boolean networks via mixed controls,” *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 254–259, 2018.
 - [74] M. Meng, G. Xiao, C. Zhai, and G. Li, “Controllability of markovian jump boolean control networks,” *Automatica*, vol. 106, pp. 70–76, 2019.
 - [75] S. Wang and H. Li, “Aggregation method to reachability and optimal control of large-size Boolean control networks,” *Science China. Information Sciences*, vol. 66, no. 7, p. 179202, 2023.
 - [76] J. Ni, H. Li, S. Zhao, Y. Guo, and D. Cheng, “Minimum-cost state-flipped control for reachability of Boolean control networks using reinforcement learning,” *IEEE Transactions on Cybernetics*, vol. 54, no. 11, pp. 7103–7115, 2024.
 - [77] Q. Zhang, J. Feng, J. Pan, and J. Xia, “Set controllability for switched Boolean control networks,” *Neurocomputing*, vol. 359, pp. 476–482, 2019.
 - [78] Q. Zhang, J. Feng, and B. Wang, “Set reachability of Markovian jump Boolean networks and its applications,” *IET Control Theory & Applications*, vol. 14, no. 18, pp. 2914–2923, 2020.
 - [79] J. Wang, Y. Liu, and H. Li, “Finite-time controllability and set controllability of impulsive probabilistic Boolean control networks,” *IEEE Access*, vol. 8, pp. 111 666–111 674, 2020.
-

- [80] Y. Li, J. Feng, X. Li, and S. Xu, "Pinning controller design for set reachability of state-dependent impulsive Boolean networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 10 838–10 850, 2022.
 - [81] W. Li and H. Li, "Set reachability and set stability of Boolean networks with state-dependent asynchronous updating rule," *Asian Journal of Control*, vol. 25, no. 6, pp. 3491–3501, 2023.
 - [82] C. Jiang, S. Fu, B. Wang, J. Zhao, and M. Sun, "Controllability and set controllability of periodically switched Boolean control networks," *International Journal of Control*, vol. 96, no. 8, pp. 2124–2132, 2023.
 - [83] X. Zhang, M. Meng, Y. Wang, and D. Cheng, "Criteria for observability and reconstructibility of Boolean control networks via set controllability," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 4, pp. 1263–1267, 2020.
 - [84] W. Liu, S. Fu, and J. Zhao, "Set stability and set stabilization of Boolean control networks avoiding undesirable set," *Mathematics*, vol. 9, no. 22, p. 2864, 2021.
 - [85] X. Zhang, Y. Wang, and D. Cheng, "Output tracking of Boolean control networks," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2730–2735, 2020.
 - [86] E. Fornasini and M. Valcher, "Feedback stabilization, regulation and optimal control of Boolean control networks," in *Proceedings of the 2014 American Control Conference*, 2014, pp. 1981–1986.
 - [87] H. Li, Y. Wang, and L. Xie, "Output tracking control of Boolean control networks via state feedback: constant reference signal case," *Automatica*, vol. 59, pp. 54–59, 2015.
 - [88] H. Li, L. Xie, and Y. Wang, "Output regulation of Boolean control networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2993–2998, 2017.
 - [89] H. Li and Y. Wang, "Output tracking of switched Boolean networks under open-loop/closed-loop switching signals," *Nonlinear Analysis: Hybrid Systems*, vol. 22, pp. 137–146, 2016.
-

- [90] Q. Zhang, J. Feng, F. Xiao, and B. Wei, "Output tracking of switched Boolean networks via self-triggered control," *IEEE Transactions on Control of Network Systems*, vol. 11, no. 3, pp. 1621–1630, 2024.
 - [91] Y. Liu, Y. Zheng, H. Li, F. Alsaadi, and B. Ahmad, "Control design for output tracking of delayed Boolean control networks," *Journal of Computational and Applied Mathematics*, vol. 327, pp. 188–195, 2018.
 - [92] S. Zhu, J. Lu, Y. Liu, T. Huang, and J. Kurths, "Output tracking of probabilistic Boolean networks by output feedback control," *Information Sciences*, vol. 483, pp. 96–105, 2019.
 - [93] Y. Wang and H. Li, "Output trackability of Boolean control networks via Ledley antecedence solution," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1183–1187, 2022.
 - [94] Y. Qiao, H. Qi, and D. Cheng, "Partition-based solutions of static logical networks with applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 1252–1262, 2018.
 - [95] D. Cheng, J. Feng, and Y. Jia, "State feedback stabilization of Boolean control networks via Ledley antecedence solution," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 145–155, 2012.
 - [96] J. Wang, H. Li, and J. Wang, "Stabilization of delayed Boolean control networks via Ledley antecedence solution," in *Proceedings of the 40th Chinese Control Conference (CCC)*, 2021, pp. 1322–1327.
 - [97] Q. Sun, H. Li, and Y. Wang, "Set stabilizability of switched Boolean control networks via Ledley antecedence solution," *Nonlinear Analysis: Hybrid Systems*, vol. 48, p. 101320, 2023.
 - [98] Y. Wang and P. Guo, "Optimal control of singular Boolean control networks via Ledley solution method," *Journal of the Franklin Institute*, vol. 356, no. 15, pp. 8948–8964, 2019.
 - [99] Y. Zhao, S. Fu, J. Zhao, and X. Li, "Robust strategy optimization of networked evolutionary games with disturbance inputs," *Journal of Systems Science and Complexity*, vol. 34, no. 1, pp. 162–174, 2021.
-

- [100] J. Kasac, B. Novakovic, D. Majetic, and D. Brezak, "Passive finite-dimensional repetitive control of robot manipulators," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 3, pp. 570–576, 2008.
 - [101] P. Mattavelli and F. Marafão, "Repetitive-based control for selective harmonic compensation in active power filters," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 5, pp. 1018–1024, 2004.
 - [102] Z. Zhang and A. Serrani, "The linear periodic output regulation problem," *Systems & Control Letters*, vol. 55, no. 6, pp. 518–529, 2006.
 - [103] C. Yuksel, S. Niculescu, and T. Vyhlídal, "A spectrum-based filter design for periodic output regulation of systems with dead-time," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 917–922, 2023.
 - [104] J. Deutscher and J. Gabriel, "Periodic output regulation for general linear heterodirectional hyperbolic systems," *Automatica*, vol. 103, pp. 136–146, 2019.
 - [105] Z. Liu and H. Li, "Identification and stabilization of faulty Boolean control networks," *Journal of the Franklin Institute*, vol. 360, no. 14, pp. 10 457–10 475, 2023.
 - [106] L. Owens, A. Benedetto, N. Dawson, C. Gaffney, and E. Parkin, "Gene therapy-mediated enhancement of protective protein expression for the treatment of Alzheimer's disease," *Brain Research*, vol. 1753, p. 147231, 2021.
 - [107] D. Cheng, F. He, H. Qi, and T. Xu, "Modeling, analysis and control of networked evolutionary games," *IEEE Transactions on Automatic Control*, vol. 60, no. 9, pp. 2402–2415, 2015.
 - [108] Y. Tang, L. Li, and J. Lu, "Modeling and optimization for networked evolutionary games with player exit mechanism: Semi-tensor product of matrices method," *Physica A: Statistical Mechanics and its Applications*, vol. 590, p. 126710, 2022.
 - [109] H. Li, Z. Geng, and M. Qin, "Strategy consensus of networked evolutionary games based on network aggregation and pinning control," *Games*, vol. 16, no. 1, p. 10, 2025.
 - [110] G. Chen and Y. Yu, "Convergence analysis and strategy control of evolutionary games with imitation rule on toroidal grid," *IEEE Transactions on Automatic Control*, vol. 68, no. 12, pp. 8185–8192, 2023.
-

- [111] G. Zhao, H. Li, P. Duan, and F. Alsaadi, “Survey on applications of semi-tensor product method in networked evolutionary games,” *Journal of Applied Analysis and Computation*, vol. 10, no. 1, pp. 32–54, 2020.
 - [112] F. Battiston *et al.*, “Networks beyond pairwise interactions: Structure and dynamics,” *Physics Reports*, vol. 874, pp. 1–92, 2020.
 - [113] —, “Higher-order interactions shape collective human behaviour,” *Nature Human Behaviour*, pp. 1–17, 2025.
 - [114] J. Guo, Y. Meng, and A. Li, “Evolutionary game dynamics for higher-order interactions,” *arXiv preprint arXiv:2501.06411*, 2025.
 - [115] J. Shi, C. Liu, and J. Liu, “Hypergraph-based model for modeling multi-agent Q-learning dynamics in public goods games,” *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 6, pp. 6169–6179, 2024.
 - [116] T. Liu, Y. Wang, and D. Cheng, “Dynamics and stability of potential hyper-networked evolutionary games,” *International Journal of Automation and Computing*, vol. 14, no. 2, pp. 229–238, 2017.
 - [117] C. Li, F. He, T. Liu, and D. Cheng, “Verification and dynamics of group-based potential games,” *IEEE Transactions on Control of Network Systems*, vol. 6, no. 1, pp. 215–224, 2018.
 - [118] J. Zhang, J. Lou, J. Qiu, and J. Lu, “Dynamics and convergence of hyper-networked evolutionary games with time delay in strategies,” *Information Sciences*, vol. 563, pp. 166–182, 2021.
 - [119] Q. Zhang, Y. Wang, Y. Wang, and Z. Xu, “Robust stability and stabilization of switched hyper-networked evolutionary games,” *IEEE Transactions on Network Science and Engineering*, vol. 12, no. 2, pp. 1420–1428, 2025.
 - [120] D. Cheng, X. Zhang, Z. Ji, and C. Li, “Hyper-networked multi-objective games,” in *Proceedings of the 44th Chinese Control Conference (CCC)*, China, 2025, pp. 8398–8403.
 - [121] P. Guo, Y. Wang, and H. Li, “Algebraic formulation and strategy optimization for a class of evolutionary networked games via semi-tensor product method,” *Automatica*, vol. 49, no. 11, pp. 3384–3389, 2013.
-

- [122] G. Zhao and Y. Wang, “Formulation and optimization control of a class of networked evolutionary games with switched topologies,” *Nonlinear Analysis: Hybrid Systems*, vol. 22, pp. 98–107, 2016.
 - [123] H. Yuan, Z. Chen, Z. Zhang, R. Zhu, and Z. Liu, “Minimum-time strategy optimization for networked evolutionary games with bankruptcy mechanism,” *Expert Systems with Applications*, vol. 237, p. 121311, 2024.
 - [124] S. Fu, Y. Pan, J. Feng, and J. Zhao, “Strategy optimisation for coupled evolutionary public good games with threshold,” *International Journal of Control*, vol. 95, no. 2, pp. 562–571, 2022.
 - [125] R. Zhu, Z. Chen, J. Zhang, and Z. Liu, “Strategy optimization of weighted networked evolutionary games with switched topologies and threshold,” *Knowledge-Based Systems*, vol. 235, p. 107644, 2022.
 - [126] S. Fu, L. Li, and J. Feng, “Strategy optimization of controlled evolutionary games on a two-layer coupled network using Lebesgue sampling,” *Nonlinear Analysis: Hybrid Systems*, vol. 56, p. 101570, 2025.
 - [127] D. Cheng, H. Qi, T. Liu, and Y. Wang, “A note on observability of Boolean control networks,” *Systems & Control Letters*, vol. 87, pp. 76–82, 2016.
 - [128] R. Albert and H. Othmer, “The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*,” *Journal of Theoretical Biology*, vol. 223, no. 1, pp. 1–18, 2003.
 - [129] C. Jiang, B. Wang, S. Fu, J. Zhao, and S. Min, “Observability of periodically switched Boolean control networks,” *Journal of Systems Science and Complexity*, vol. 36, no. 3, pp. 985–1001, 2023.
 - [130] A. Civilini, O. Sadekar, F. Battiston, J. Gómez-Gardeñes, and V. Latora, “Explosive cooperation in social dilemmas on higher-order networks,” *Physical Review Letters*, vol. 132, no. 16, p. 167401, 2024.
-