Input-State Approach to Boolean Networks

Daizhan Cheng, *Fellow, IEEE*

***Abstract*—This paper investigates the structure of Boolean networks via input-state structure. Using the algebraic form proposed by the author, the logic-based input-state dynamics of Boolean networks, called the Boolean control networks, is converted into an algebraic discrete-time dynamic system. Then the structure of cycles of Boolean control systems is obtained as compounded cycles. Using the obtained input-state description, the structure of Boolean networks is investigated, and their attractors are revealed as nested compounded cycles, called *rolling gears*. This structure explains why small cycles mainly decide the behaviors of cellular networks. Some illustrative examples are presented.**

***Index Terms*—Algebraic form, input-state structure, invariant subspace, network transition matrix.**

I. Introduction

INPUT–OUTPUT structure is essential in systems and control theory. How about the cellular networks? It was pointed out by [11] that "Gene-regulatory networks are defined by trans and cis logic. ... Both of these types of regulatory networks have input and output." Ignoring outputs, this paper focuses on input-state structure only.

A Boolean network could be a description of genetic circuits, an explanation of self-organization in organisms, and the structure causing order in the evolution, which leads to life [12]. In Boolean network model, gene expression is quantized to only two levels: "T"(True) and "F"(False), or "1" and "0," respectively, denoted by $D = \{T, F\}$, (or $D = \{1, 0\}$). We refer to [7] for logical notations, concepts, and operators used in this paper, and refer to [2] for some related works in neural networks.

Denote the nodes of a network graph by $A_1(t), A_2(t), \ldots$. Each node is functionally related to the expression states of some other nodes. If A_i is affected directly by A_j, there is a directed edge from A_j to A_i, and it is said that A_j is in the neighborhood of A_i. It can also be understood as the intracellular signal transduction from jth cellular to ith cellular. Throughout this paper, we consider only the networks that have fixed graph topologies. The actions between genes are described by logical rules, which are described by a logical dynamic equation [6]

$$\begin{cases} A_1(t+1) = f_1(A_1(t), A_2(t), \ldots, A_n(t)) \\ A_2(t+1) = f_2(A_1(t), A_2(t), \ldots, A_n(t)) \\ \vdots \\ A_n(t+1) = f_n(A_1(t), A_2(t), \ldots, A_n(t)) \end{cases} \tag{1}$$

Manuscript received May 11, 2008; revised November 04, 2008; accepted December 02, 2008. Current version published February 27, 2009. This work was supported in part by the National Natural Science Foundation (NNSF) of China under Grants 60674022, 60736022, and 60221301.

The author is with the Institute of Systems Science, Chinese Academy of Sciences, Beijing 100190, China (e-mail: dcheng@iss.ac.cn).

Digital Object Identifier 10.1109/TNN.2008.2011359

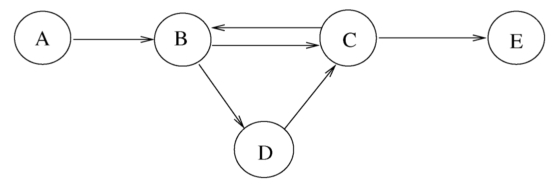

Fig. 1. Graph of network.

where f_i, $i = 1, 2, \ldots, n$, are logical functions (also called n-ary logical operators[1]).

We use an example to illustrate the graph and dynamics of a network. It will be used again in the sequel.

Example 1.1: The graph of a Boolean network is depicted in Fig. 1.

Its dynamic model is assumed to be

$$\begin{cases} A(t+1) = A(t) \\ B(t+1) = A(t) \to C(t) \\ C(t+1) = B(t) \vee D(t) \\ D(t+1) = \neg B(t) \\ E(t+1) = \neg C(t). \end{cases} \tag{2}$$

If the number of nodes in a Boolean network is n, then it is obvious that the state space considers of 2^n statues, which is a finite set described as D^n. So as a dynamic process on D^n, there must be at least a fixed point or a cycle, and eventually a trajectory starting from any initial state must enter a cycle (a fixed point can be considered as a cycle of length one). So a cycle is also called an *attractor*. For convenience, we briefly denote by Ω the set of attractors. For a state x_0, the smallest number of steps to enter Ω is called its *transient period*, denoted by $T_t(x_0)$. That is, let $x(t, x_0)$ be the trajectory starting from x_0 (i.e., $x(0, x_0) = x_0$). Then

$$T_t(x_0) = \min\{k \mid x(k, x_0) \in \Omega\}.$$

For overall network, the transient period is defined as

$$T_t = \max_{x \in D^n} T_t(x).$$

One of the most important issues in investigating a Boolean network is to find its cycles and transient period. These topics have been studied widely, e.g., in [6] and [10]. But there was no reported technique, which solves the problem systematically so far.

To investigate the structure of a Boolean network, Cheng and Qi [5] proposed a way to convert system (1) into a standard discrete-time dynamic system. The key tool for this approach is the matrix expression of logic, based on semitensor product of matrices. We give a very brief introduction here and refer to

[3] and [4] for details. Investigating the structure of a network via its dynamics can also been found in [8], [13], and [16].

Definition 1.2:

1) Let X be a row vector of dimension np, and Y be a column vector with dimension p. Then we split X into p equal-size blocks as $X^1, \ldots, X^p$, which are $1 \times n$ rows. Define the semitensor product (STP), denoted by $\ltimes$, as

$$\begin{cases} X \ltimes Y = \sum\limits_{i=1}^{p} X^i y_i \in \mathbb{R}^n \\ Y^T \ltimes X^T = \sum\limits_{i=1}^{p} y_i (X^i)^T \in \mathbb{R}^n. \end{cases} \tag{3}$$

2) Let $A \in M_{m\times n}$ and $B \in M_{p\times q}$. If either n is a factor of p, say $nt = p$, denoted as $A \prec_t B$, or p is a factor of n, say $n = pt$, denoted as $A \succ_t B$, then we define the STP of A and B, denoted by $C = A \ltimes B$, as the following: C consists of $m \times q$ blocks as $C = (C^{ij})$ and each block is

$$C^{ij} = A^i \ltimes B_j, \qquad i = 1, \ldots, m, \qquad j = 1, \ldots, q$$

where A^i is ith row of A and B_j is the jth column of B.

STP of matrices is a generalization of the conventional matrix product, so the notation $\ltimes$ can be omitted. Moreover, all the basic properties of the conventional matrix product remain true for this extension.

Set

$$\Delta_k := \{\delta_k^i \mid i = 1, 2, \ldots, k\}$$

where δ_k^i is the ith column of identity matrix I_k. In the framework of STP, the logical variables are expressed as a vector in Δ_2 by identifying T with $[1, 0]^T$ and F with $[0, 1]^T$. Then the region D is replaced by

$$D_v = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}.$$

In this way, for a logical operator, we can always find a matrix, called its structure matrix, and the action of a logical operator on logical variables becomes a (semitensor) product of the structure matrix with its arguments' vectors. For instance, consider conjunction "$\wedge$" [7], then its structure matrix is

$$M_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

Hence, (in vector form) we have

$$P \wedge Q = M_c P Q, \qquad P, Q \in D_v.$$

In general, we have the following proposition.

Proposition 1.3: Let f be a logical function (operator) of $X_1, X_2, \ldots, X_n$, and

$$Y = f(X_1, X_2, \ldots, X_n). \tag{4}$$

Then we can find a matrix, called the structure matrix of f and denoted by M_f, such that

$$Y = M_f X_1 X_2 \cdots X_n. \tag{5}$$

Note that X_i may appear in (4) many times, but in (5), Y is multilinear with respect to $X_1, \ldots, X_n$.

Using this vector expression, Cheng and Qi [5] convert a logical equation of a Boolean network into a discrete time linear system, called its algebraic form as $x(t+1) = Lx(t)$. Analyzing the structure of L, precise formulas have been obtained to reveal the structure of the network.

This paper considers the Boolean control networks, which have input-state structure. We first propose a framework for Boolean control networks, and the structure of attractors of the networks is investigated. The input-state approach is then applied to the analysis of the structure of general Boolean networks and a structure of *nested compounded cycles* is obtained. We call such a structure "rolling gears" structure and will discuss some interesting properties of this kind of structures. We guess it could be used to reveal the *hidden order* in lives.

This paper is organized as follows. Section II reviews the converting technique from logical dynamic equation to algebraic one. An example is used to depict it. In Section III, a framework of Boolean control networks is proposed and the structure of attractors of input-state type of networks is investigated. In Section IV, the input-state approach is implemented to analyze the structure of general Boolean networks and the structure of nested compounded cycles, called "rolling gears," is revealed. Section V contains two illustrative examples. Section VI is a brief conclusion.

II. From Logical Equation to Algebraic Equation

In this section, we briefly review the technique, developed in [5], which provides a systematic tool to treat Boolean networks. Assume a logical variable is expressed in a vector form. That is, $A_i(t) \in D_v \sim D$. (We will use scalar form D and vector form D_v alternatively without explanation, and use D for both. From the text, it is very easy to figure out what form is used there.) Consider system (1). Since $f_i, i = 1, 2, \ldots, n$, are logical functions, according to Proposition 1.3, we can convert (1) into an algebraic dynamic form as

$$\begin{cases} A_1(t+1) = M_1 A_1(t) A_2(t) \cdots A_n(t) \\ A_2(t+1) = M_2 A_1(t) A_2(t) \cdots A_n(t) \\ \vdots \\ A_n(t+1) = M_n A_1(t) A_2(t) \cdots A_n(t) \end{cases} \tag{6}$$

where $M_i, i = 1, \ldots, n$, are the structure matrices of f_i. Define

$$x(t) = A_1(t) A_2(t) \cdots A_n(t) \in \Delta_{2^n}. \tag{7}$$

Multiplying $A_i(t+1)$ together yields

$$x(t+1) = \prod_{i=1}^{n} [M_i A_1(t) A_2(t) \cdots A_n(t)]. \tag{8}$$

Using the properties of STP of matrices, (8) can be converted into an algebraic form as

$$x(t+1) = Lx(t) \tag{9}$$

where L is called the *network transition matrix* of (1).

The following result reveals all the attractors from L.

Theorem 2.1: Consider system (1) with its network transition matrix L.

1) The number of length s cycles N_s is inductively determined by

$$\begin{cases} N_1 = \mathrm{Trace}(L) \\ N_s = \dfrac{\mathrm{Trace}(L^s) - \sum_{k\in\mathcal{P}(s)} kN_k}{s}, \qquad 2 \le s \le 2^n \end{cases} \tag{10}$$

where $\mathcal{P}(s)$ is the set of proper factors of s.

2) The elements on cycles of length s, denoted by $\mathcal{C}_s$, is

$$\mathcal{C}_s = \mathcal{D}_a(L^s) \backslash \cup_{k\in\mathcal{P}(s)} \mathcal{D}_a(L^k) \tag{11}$$

where $\mathcal{D}_a(L)$ is the set of diagonal nonzero columns of L.

Note that $a \in \mathcal{P}(k)$, iff $a \in \mathbb{Z}_+$, $a < k$, and $k/a \in \mathbb{Z}_+$. For instance, $\mathcal{P}(8) = \{1, 2, 4\}$, $\mathcal{P}(12) = \{1, 2, 3, 4, 6\}$, etc.

Denote

$$r_0 = \min\left\{k \,\middle|\, L^k \in \{L^{k+1}, L^{k+2}, \ldots, L^{2^{2n}}\}\right\}.$$

Then, we have the following theorem.

Theorem 2.2: For system (1), the transient period

$$T_t = r_0 = \min\left\{k \,\middle|\, L^k \in \{L^{k+1}, L^{k+2}, \ldots, L^{2^{2n}}\}\right\}. \tag{12}$$

The following theorem provides an easy way to construct the regions of attraction.

Theorem 2.3: Given an $\eta \in \Delta_{2^n}$, denote the columns of L, which equal to η, by $L_{i_j} = \eta$, $j = 1, 2, \ldots, k$. Then the set of parent points of η is

$$L^{-1}(\eta) = \left\{\delta_{2^n}^{i_1}, \delta_{2^n}^{i_2}, \ldots, \delta_{2^n}^{i_k}\right\}. \tag{13}$$

We use an example to illustrate these results.

Example 2.4: Consider Example 1.1. Equation (2) can be converted into algebraic form as

$$\begin{cases} A(t+1) = A(t) \\ B(t+1) = M_i A(t) C(t) \\ C(t+1) = M_d B(t) D(t) \\ D(t+1) = M_n B(t) \\ E(t+1) = M_n C(t). \end{cases} \tag{14}$$

Let $x(t) := A(t)B(t)C(t)D(t)E(t)$. Then

$$\begin{aligned} x(t+1) &= A(t)M_iA(t)C(t)M_dB(t)D(t)M_nB(t)M_nC(t) \\ &= (I_2 \otimes M_i)M_rACM_dBDM_nBM_nC \\ &= \ldots(I_4 \otimes M_d)ACBDM_nBM_nC \\ &= \ldots(I_{16} \otimes M_n)ACBDBM_nC \\ &= \ldots ACBW_{[2]}BDM_nC \\ &= \ldots(I_8 \otimes W_{[2]})ACB^2DM_nC \\ &= \ldots(I_4 \otimes M_r)(I_{16} \otimes M_n)ACBDC \\ &= \ldots AW_{[2]}BCW_{[2]}CD \\ &= \ldots(I_2 \otimes W_{[2]})(I_8 \otimes W_{[2]})ABC^2D \\ &= \ldots(I_4 \otimes M_r)A(t)B(t)C(t)D(t). \end{aligned} \tag{15}$$

Starting from the second row of (15), the front constant coefficient matrix in the previous row is replaced in the next row by "$\ldots$" to save space.

Note that now in the left-hand side of (15) there is no $E(t)$. To get $x(t)$ we have to add it. We can use a dummy operator [5]

$$E_d = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

which satisfies

$$E_dPQ = Q \qquad \forall P, Q \in \Delta_2.$$

Hence, (15) can be converted further as

$$\begin{aligned} x(t+1) &= \ldots(I_4 \otimes M_r)A(t)B(t)C(t)E_dW_{[2]}D(t)E(t) \\ &= \ldots(I_8 \otimes (E_dW_{[2]}))x(t). \end{aligned} \tag{16}$$

From (16), we have

$$\begin{aligned} L = (I_2 \otimes M_i)M_r(I_4 \otimes M_d)(I_{16} \otimes M_n)(I_8 \otimes W_{[2]}) \\ (I_4 \otimes M_r)(I_{16} \otimes M_n)(I_2 \otimes W_{[2]})(I_8 \otimes W_{[2]}) \\ (I_4 \otimes M_r)(I_8 \otimes (E_dW_{[2]})). \end{aligned} \tag{17}$$

Note that $L \in M_{32\times 32}$. It is easy to calculate L. We express it into a *condensed form* as

$$\begin{aligned} L = \delta_{32}[&4, 4, 4, 4, 11, 11, 11, 11, 2, 2, 6, 6, 9, 9, 13, 13, 20, 20, \\ &20, 20, 19, 19, 19, 19, 18, 18, 22, 22, 17, 17, 21, 21] \end{aligned}$$

where the ith component μ means the ith column of L is δ_{32}^{μ}.

It is easy to calculate that only when $i = 1$, or 2, $N_i \neq 0$, and $\mathrm{Trace}(L) = 2$ and $\mathrm{Trace}(L^2) = 4$. Theorem 2.1 yields that there are two fixed points, which are

$$11100 \quad \text{and} \quad 01100$$

and a cycle of length 2, which is

$$11010 \rightarrow 10101 \rightarrow 11010.$$

We can also check that $r_0 = 4$ and $L^4 = L^6$, so the transient period is $T_t = 4$.

III. Boolean Control Networks

A Boolean control network is defined as

$$\begin{cases} A_1(t+1) = f_1(A_1(t), A_2(t), \ldots, A_n(t), u_1(t), \ldots, u_m(t)) \\ A_2(t+1) = f_2(A_1(t), A_2(t), \ldots, A_n(t), u_1(t), \ldots, u_m(t)) \\ \vdots \\ A_n(t+1) = f_N(A_1(t), A_2(t), \ldots, A_n(t), u_1(t), \ldots, u_m(t)) \end{cases} \tag{18}$$

where u_i, $i = 1, 2, \ldots m$, are inputs (or controls), which are logical variables satisfying certain logical rule, called the input network, described as

$$\begin{cases} u_1(t+1) = g_1(u_1(t), u_2(t), \ldots, u_m(t)) \\ u_2(t+1) = g_2(u_1(t), u_2(t), \ldots, u_m(t)) \\ \vdots \\ u_m(t+1) = g_m(u_1(t), u_2(t), \ldots, u_m(t)). \end{cases} \tag{19}$$

In an algebraic form, a Boolean control network can be expressed as

$$\begin{cases} u(t+1) = Gu(t), & u \in D^m \\ x(t+1) = L(u)x(t), & x \in D^n \end{cases} \tag{20}$$

where $L(u) = Lu(t)$ is the control-depending network transition matrix.

Example 3.1: Consider Example 1.1 again. It is very natural to take $A(t)$ as input. Ignoring $E(t)$, which is considered as an output, the system can be rewritten as

$$\begin{cases} B(t+1) = u(t) \to C(t) \\ C(t+1) = B(t) \vee D(t) \\ D(t+1) = \neg B(t) \end{cases} \tag{21}$$

and the control network is

$$u(t+1) := A(t+1) = A(t).$$

Converting this system into an algebraic form, we have

$$\begin{cases} u(t+1) = u(t) \\ x(t+1) = L(u)x(t) \end{cases} \tag{22}$$

where $L(u)$ can be easily calculated as

$$L(u) = M_c u(I_2 \otimes Md)(I_8 \otimes Mn)W_{[2]}W_{[2,8]}M_r$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1-\alpha & 0 \\ 1 & 1 & 1-\alpha & 1-\alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1-\alpha \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha & 0 \\ 0 & 0 & \alpha & \alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and $u = (\alpha, 1-\alpha)^T$, where $\alpha = 0$ or 1.

Now both δ_2^1 and δ_2^2 are fixed points of the control network. Using Theorem 2.1, it is easy to figure out that for $u = \delta_2^1$, there is a fixed point for the system, which is $x = (01000000) \sim 110$, and there is also a cycle of length 2, which is $101 \to 010 \to 101$. While $u = \delta_2^2$, there is only a fixed point 110.

In general, we consider the structure of the Boolean control system (18), where the controls are varying, according to its own logical evolution rule (19).

Denote by $U = D^m$ the input space, by $X = D^n$ the state space, and let $W = U \times X$ be the input-state (product) space. Let $w \in W$. It is easy to prove that there exist unique $u \in U$ and $x \in X$, such that $w = ux$. Now assume there is a cycle of length k in the input-state space W. Say, it is

$$\begin{aligned} C_w^k : w(0) &= w_0 = u_0x_0 \to w(1) = w_1 \\ &= u_1x_1 \to \cdots \to w(k) = w_k = u_kx_k = w_0. \end{aligned}$$

First, one sees easily that since $u_0 = u_k$, $\{u_0, u_1, \ldots, u_k\}$ contains, say, j folds of a cycle of length ℓ, say, $j\ell = k$. Hence $u_\ell = u_0$. Now let us see what condition the $\{x_i\}$ in the cycle C_w^k should satisfy. Define a network transition matrix as

$$\Psi := L(u_{\ell-1})L(u_{\ell-2})\ldots L(u_1)L(u_0). \tag{23}$$

Starting from $w_0 = u_0x_0$, we have x component of the cycle C_w^k as

$$\begin{aligned} x_0 \to x_1 &= L(u_0)x_0 \to x_2 = L(u_1)L(u_0)x_0 \\ \to \cdots \to x_\ell &= \Psi x_0 \to x_{\ell+1} = L(u_0)\Psi x_0 \\ \to x_{\ell+2} &= L(u_1)L(u_0)\Psi x_0 \to \cdots \to x_{2\ell} = \Psi^2 x_0 \to \\ &\vdots \\ x_{(j-1)\ell+1} &= L(u_0)\Psi^{j-1}x_0 \to x_{(j-1)\ell+2} \\ &= L(u_1)L(u_0)\Psi^{j-1}x_0 \\ \to \cdots \to x_{j\ell} &= \Psi^j x_0 = x_0. \end{aligned} \tag{24}$$

We conclude that $x_0 \in D^n$ is a fixed point of the equation (with $j > 0$ being the smallest one)

$$x(t+1) = \Psi^j x(t). \tag{25}$$

Conversely, if $x_0 \in D^n$ is a fixed point of (25) and u_0 is a point on a cycle of control space C_u^ℓ, then it is obvious that we have the cycle (24).

Summarizing above arguments yields the following theorem.

Theorem 3.2: Consider the Boolean control network (20). A set $C_w^k \subset D^{k(n+m)}$ is a cycle of the control system with length k, iff for any point $w_0 = u_0x_0 \in C_w^k$, there exists an $\ell \le k$ as a factor of k, such that $u_0, u_1 = Gu_0, u_2 = G^2u_0, \ldots, u_\ell = G^\ell u_0 = u_0$ is a cycle in the control space, and x_0 is a fixed point of (25) (with $j > 0$ being the smallest one).

Theorem 3.2 shows how to find all the cycles in the input-state space. First, we can find the cycles in the input space. Pick a cycle in the input space, say C_u^ℓ, then for each point $u_0 \in C_u^\ell$, we can construct an auxiliary system

$$x(t+1) = \Psi x(t). \tag{26}$$

Now, say $C_u^\ell = (u_0, u_1, \ldots, u_\ell = u_0)$ is a cycle in U, and $C_x^j = (x_0, x_1, \ldots, x_j = x_0)$ is a cycle of (26). Then, a cycle C_w^k, $k = \ell j$, can be constructed by

$$\begin{aligned} w_0 = u_0x_0 \to w_1 &= u_1L(u_0)x_0 \to w_2 = u_2L(u_1)L(u_0)x_0 \\ \to \cdots \to w_\ell &= u_0x_1 \to w_{\ell+1} = u_1L(u_0)x_1 \\ \to w_{\ell+2} &= u_2L(u_1)L(u_0)x_1 \to \cdots \to \\ &\vdots \\ \to w_{(j-1)\ell} &= u_0x_{(j-1)} \to w_{(j-1)\ell+1} = u_1L(u_0)x_{(j-1)} \\ \to w_{(j-1)\ell+2} &= u_2L(u_1)L(u_0)x_{(j-1)} \to \cdots \to \\ \to w_{j\ell} &= u_0x_j = u_0x_0 = w_0. \end{aligned} \tag{27}$$

We call this C_w^k the *compounded cycle* of C_u^ℓ and C_x^j, denoted by $C_w^k = C_u^\ell \circ C_x^j$.

Note that from a cycle in the input space C_u^ℓ, we can choose any point as the starting point u_0. Then, in (26), we have different Ψ, which produces different C_x^j. It is reasonable to guess that the final $C_w^k = C_u^\ell \circ C_x^j$ is independent of the choice of u_0. Otherwise, the picture will be incorrect. In the following, we will prove this is true.

Definition 3.3: Let $C_w^k = \{w(t) | t = 0, 1, \ldots, k\}$ be a cycle in the input-state space, and C_u^ℓ be a cycle in the input space.

Splitting $w(t) = u(t)x(t)$, we said that C_w^k is attached to C_u^ℓ at u_0, if $w(0) = u_0 x_0$, and

1) $u(t) \in C_u^\ell$, with $u(0) = u_0$;
2) $x(0) = x_0$ is a fixed point of (25) with $j = k/\ell \in \mathbb{Z}_+$.

Remark 3.4: According to Theorem 3.2, each cycle C_w^k in the input-state space must be attached precisely to one cycle in the input space. In fact, the following argument shows that C_w^k attaches C_u^ℓ at u_0 at moment $t = 0$ (and the attaching point of C_w^k is $w_0 = u_0 x_0$) and will attach it at u_1 at moment $t = 1$ (with the attaching point of C_w^k being $w_1 = u_1 x_1$) and so on. So C_w^k and C_u^ℓ are moving as two assembled gears.

Proposition 3.5: The sets of the cycles in the input-state space, attached to any point of a given cycle C_u^ℓ, are the same.

Proof: Let $C_u^\ell = \{u_0, u_1, \ldots, u_\ell = u_0\}$ be the cycle we are concerned with. Let $S_0, S_1, \ldots, S_{\ell-1}$ be the set of cycles attached to $u_0, u_1, \ldots, u_{\ell-1}$, respectively. First, we show that

$$S_0 \subset S_i, \qquad i = 1, 2, \ldots, \ell - 1.$$

Let $C_k^0 = \{w_0, w_1, \ldots, w_k\} \in S_0$, i.e., it is a cycle attached to C_u^ℓ at u_0. Using the elements of a control cycle, we can define

$$L_i := L(u_i), \qquad i = 0, 1, \ldots, \ell - 1.$$

Then, we construct ℓ system matrices as

$$\begin{cases} \Psi_0 := L_{\ell-1}L_{\ell-2}\ldots L_0, \\ \Psi_1 := L_0 L_{\ell-1}L_{\ell-2}\ldots L_1, \\ \vdots \\ \Psi_{\ell-1} := L_{\ell-2}L_{\ell-3}\ldots L_0, L_{\ell-1}. \end{cases}$$

Correspondingly, we then construct ℓ auxiliary systems as

$$x(t+1) = \Psi_i x(t), \qquad i = 0, 1, \ldots, \ell - 1. \tag{28}$$

Since $w_0 = u_0 x_0 \in C_w^k \in S_0$, then x_0 satisfies

$$(\Psi_0)^j x_0 = x_0. \tag{29}$$

Note that $w(1) := w_1 = u_1 L_0 x_0$. To see that $w_1 \in C_k^1 \in S_1$, we have to show that $L_0 x_0$ satisfies

$$(\Psi_1)^j L_0 x_0 = L_0 x_0. \tag{30}$$

This is true because

$$\begin{aligned} L_0 x_0 &= L_0 (\Psi_0)^j x_0 = L_0 \left(L_{\ell-1}\ldots L_0\right)^j x_0 \\ &= L_0 \underbrace{(L_{\ell-1}\ldots L_0)\ldots(L_{\ell-1}\ldots L_0)}_{j} x_0 \\ &= \underbrace{(L_0 L_{\ell-1}\ldots L_1)\ldots(L_0 L_{\ell-1}\ldots L_1)}_{j} L_0 x_0 \\ &= (L_0 L_{\ell-1}\ldots L_1)^j L_0 x_0 = \Psi_1^j L_0 x_0. \end{aligned}$$

Similarly, we can show that

$$u_s L_{s-1} L_{s-2} \ldots L_0 x_0 \in C_k^s \in S_s, \qquad s = 1, 2, \ldots, \ell - 1.$$

Note that, precisely speaking, (30) can only assure that there is a cycle of length $\ell \times j'$ attached to the cycle at u_1, where j' is a factor of j. But since the above definition of $\{\Psi_i\}$ is on a rotating style, starting from a point $w_0 = u_1 x_0'$, the same argument shows $j \le j'$. So $j' = j$. ∎

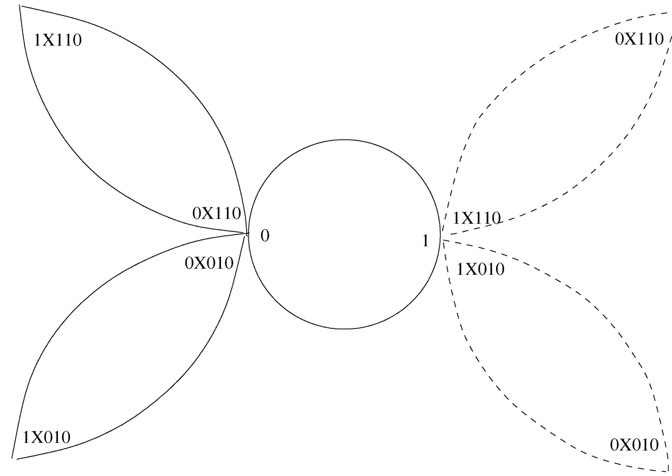

Fig. 2. Circles of a control system.

Example 3.6: Revisit Example 3.1. Now we may change the control to

$$A(t+1) = \neg A(t).$$

We have an obvious control cycle: $0 \to 1 \to 0$. Then, we can easily calculate (using condensed form)

$$\begin{aligned} L(0) &= \delta_8(2,2,2,2,1,3,1,3) \\ L(1) &= \delta_8(2,2,6,6,1,3,5,7). \end{aligned}$$

Then, we consider auxiliary system

$$x(t+1) = \Psi x(t) \tag{31}$$

where

$$\Psi = L(1)L(0) = \delta_8(2,2,2,2,2,6,2,6).$$

The routine calculation shows the following: nontrivial power of Ψ is 1 and $\mathrm{Trace}(\Psi^1) = 2$. So there are two fixed points, which are $(0,1,0,0,0,0,0,0) \sim 110$ and $(0,0,0,0,0,0,1,0) \sim 010$. The overall cycles are depicted in Fig. 2, where the dash lines show the duplicated cycles. Overall, we have a cycle in the input space and two compounded cycles of length 2 in the input-state space. Fig. 2 shows the cycles of this control network.

Finally, we consider the transient period of compounded cycles. Assume C_u^i, $i = 1, \ldots, p$, are the cycles in control space. For a fixed C_u^i, which has length ℓ_i, we can construct Ψ_i and find the smallest r^i such that

$$(\Psi_i)^{r^i} = (\Psi_i)^{r^i + T_i}$$

then it is clear that if a point will eventually enter the cycles attached to this cycle, then after r^i (compounded) steps the second component will enter the rotating cycle. Note that Ψ_i is a compounded mapping, consisting of ℓ_i steps. Taking the first part (C_u^ℓ) into consideration, it is easily seen that the transient period for cycles attached to C_u^i, denoted by $T_t(C_u^i)$, satisfies

$$\max\left\{r_0, \ell_i(r^i - 1)\right\} \le T_t(C_u^i) \le \max\left\{r_0, \ell_i(r^i)\right\} \tag{32}$$

where $i = 1, \ldots, p$.

Define

$$V_i := \max\left\{r_0, \ell_i(r^i - 1)\right\}$$
$$U_i := \max\left\{r_0, \ell_i(r^i)\right\}, \qquad i = 1, \dots, p.$$

Then, the following is obvious.

Proposition 3.7: The transient period of the control network satisfies

$$\max_{1\le i\le p}\{V_i\} < T_t \le \max_{1\le i\le p}\{U_i\}. \tag{33}$$

IV. Cascaded Boolean Networks

The input-state structure proposed in the previous section is very useful in analyzing the structure of Boolean networks with the cascading structure.

Definition 4.1: Consider system (1), where $x \in X = D^n$. A subspace $V = D^k \subset X$ is called an invariant subspace, if $x_0 \in V$ implies $x(t, x_0) \in V, \forall t > 0$.

From Section III, one sees easily that the control space is an invariant subspace of the control-state (product) space. Conversely, an invariant subspace can also be considered as a control space.

To testify if a subspace is a control invariant subspace, we can use either network graph or network equation. We use the following two examples to illustrate this.

Let $\{A_{i_1}, \dots, A_{i_s}\}$ be a subset of the nodes of a network. $V = \mathrm{Span}\{A_{i_1}, \dots, A_{i_s}\}$ means that $V \subset X$ is the subspace of the states of the subnetwork with nodes $\{A_{i_1}, \dots, A_{i_s}\}$ and edges between them, inherited from original graph.

Example 4.2: Consider Fig. 3. One sees easily that $V_1 = \mathrm{Span}\{A\}$ and $V_2 = \mathrm{Span}\{A, B, C, D\}$ are two invariant subspaces. We have the nested invariant subspaces as

$$V_1 \subset V_2 \subset X.$$

Note that $V = \mathrm{Span}\{A, B, C\}$ is not an invariant subspace, because it will be affected by D. (If you are familiar with graph theory, it is easy to see that a subspace is invariant iff the subgraph has in-degree zero.)

The structure of nested invariant subspaces can also be discovered from network equations. Consider the following example.

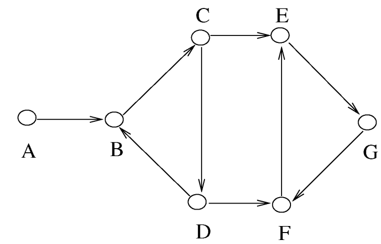

Fig. 3. Invariant subspaces.

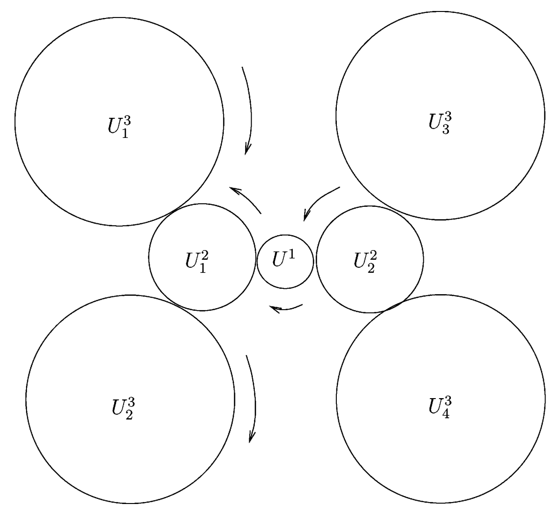

Fig. 4. Structure of cycles in a cascaded Boolean network.

Example 4.3: Consider the system shown in (34) at the bottom of the page. Then we have at least two nested invariant subspaces: $V_1 = \mathrm{Span}\{A_1, \dots, A_\ell\} = D^\ell$ and $V_2 = \mathrm{Span}\{A_1, \dots, A_\ell, B_1, \dots, B_m\} = D^{\ell+m}$, and

$$V_1 \subset V_2 \subset X = D^{\ell+m+n}.$$

We consider a cycle, say, $U^3 \in X$. As we discussed in Section III, it must be attached to a cycle, say, $U^2 \in V_2$. Similarly, U_2 must attach to a cycle, say, $U^1 \in V_1$. Now in Fig. 4, we assume that two cycles $U_1^2, U_2^2 \in V_2$ are attached to $U_1 \in V_1$, $U_1^3, U_2^3 \in X$ are attached to U_1^2, and $U_3^3, U_4^3 \in X$ are attached to U_2^2. We call such connected cycles chaired gears.

Chaired gears have the following properties.

- Each chair of gears, such as $U^1 \to U_1^2 \to U_1^3$, has multiplicative perimeters (precisely, the numbers of states in

$$\begin{cases} A_1(t+1) = f_1^1(A_1(t), \dots, A_\ell(t)) \\ \vdots \\ A_\ell(t+1)) = f_\ell^1(A_1(t), \dots, A_\ell(t)) \\ B_1(t+1) = f_1^2(A_1(t), \dots, A_\ell(t), B_1(t), \dots, B_m(t)) \\ \vdots \\ B_m(t+1)) = f_m^2(A_1(t), \dots, A_\ell(t), B_1(t), \dots, B_m(t)) \\ C_1(t+1) = f_1^3(A_1(t), \dots, A_\ell(t), B_1(t), \dots, B_m(t), C_1(t), \dots, C_n(t)) \\ \vdots \\ C_n(t+1)) = f_n^3(A_1(t), \dots, A_\ell(t), B_1(t), \dots, B_m(t), C_1(t), \dots, C_n(t)). \end{cases} \tag{34}$$

TABLE I
TRUTH TABLE OF (35)

	f_1	f_2	f_3	f_4	f_5
	1	1	1	1	1
	1	1	0	1	0
	1	1	1	1	0
	0	0	1	0	0
	1	0	0	1	0
	1	1	1	0	0
	1	1	1	0	0
	0	0	0	0	0
j_1	5	3	3	3	5
j_2	2	5	1	4	4
j_3	4	4	5	4	1

cycles), i.e., the perimeter of U_1^3 is a multiplier of the perimeter of U_1^2, and the perimeter of U_1^2 is a multiplier of the perimeter of U^1.

- In each chair, the smaller gears affect the larger gears, and the larger gears do not affect the smaller gears.
- Smallest gears look like steering gears, which steer the other gears to run.

Kauffman claimed that [12] in a cellular network the tiny attractors decide the vast order. The "rolling gears" structure explained why small cycles decide the order of the whole network. We guess that the structure of "rolling gears" may be used to explain the "hidden order" in lives.

Finally, one may ask: Why should there be an invariant subspace? In fact, if a large or huge network has small cycles, then the small cycles with the elements in their region of attraction form small invariant subspaces. If there are no such small cycles, the system is in chaos [12]. So an ordered large scale network should have the structure of nested invariant subspaces.

V. TWO ILLUSTRATIVE EXAMPLES

First example is from [14]. It is used for two purposes: 1) showing the standard algorithm; and 2) demonstrating that the "small cycles" have decisive importance for the structure of the overall network.

Example 5.1: Consider a system with five nodes, as

$$A_i = f_i(A_{j_1}, A_{j_2}, A_{j_3}), \qquad i = 1,2,3,4,5 \tag{35}$$

where the logical functions f_i, $i = 1, \dots, 5$, are determined by the truth table (Table I).

Then, the matrix form of system (35) is

$$\begin{cases} A(t+1) = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} E(t)B(t)D(t) \\ B(t+1) = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} C(t)E(t)D(t) \\ C(t+1) = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} C(t)A(t)E(t) \\ D(t+1) = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} C(t)D(t)D(t) \\ E(t+1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} E(t)D(t)A(t). \end{cases} \tag{36}$$

To get the structure matrix, note that the first row of the structure matrix of f_i is exactly the same as its values in truth table.

To convert the matrix form back to logical form, $\mathrm{mod}(2)$ algebra is more convenient. Using $\mathrm{mod}(2)$ algebra, system (35) can be expressed as

$$\begin{cases} A(t+1) = B(t) +_2 D(t) +_2 B(t) \times_2 D(t) \\ B(t+1) = D(t) +_2 E(t) +_2 C(t) \times_2 D(t) \times_2 E(t) \\ C(t+1) = A(t) +_2 C(t) +_2 E(t) +_2 C(t) \times_2 E(t) \\ +_2 A(t) \times_2 C(t) \times_2 E(t) \\ D(t+1) = D(t) \\ E(t+1) = A(t) \times_2 D(t) \times_2 E(t). \end{cases} \tag{37}$$

It is easy to see that the structure matrix of $\mathrm{mod}(2)$ times $\times_2$ is M_c and $\mathrm{mod}(2)$ plus "$+_2$" is

$$M_+ = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

Let $x(t) = A(t)B(t)C(t)D(t)E(t)$. Then

$$x(t+1) = M_+^2 BDM_cBDM_+^2DEM_c^2CDE \\ M_+^4ACEM_cCEM_c^2ACEDM_c^2ADE.$$

Now there is a normal routine to figure out L. In fact

$$L = \delta_{32}[\,1,6,4,16,13,2,8,12,1,6,20,32,13,2,24,28, \\ 2,2,4,12,10,6,4,16,2,2,20,28,10,6,20,32\,].$$

Then, one can check that the nontrivial powers are 1 and 2, and

$$\mathrm{Trace}(L) = 4 \quad \mathrm{Trace}(L^2) = 6.$$

We conclude that there are four fixed points and one cycle of length 2. Using Theorem 2.1, one sees easily that the fixed points are

$$E_1 = 11111 \quad E_2 = 10011 \quad E_3 = 00100 \quad E_4 = 00000$$

and the cycle of length 2 is

$$11110 \to 11010 \to 11110.$$

The smallest repeating L^k is $L^3 = L^5$, so the transient period $T_t = 3$.

Finally, we use Theorem 2.3 to get the whole picture of the state space.

- Starting from $E_1 = 11111$, we calculate its parent states, its grand parent states, and so on. We have (in the following, $[x]$ is used to show that x is already on the cycle, so we remove it from the retrieving chain)
$$E_1 = 11111 \sim \delta^1 \Rightarrow [L_1 \to \delta^1] \\ L_9 \to \delta^9 \sim 10111 \Rightarrow \emptyset.$$
- $$E_2 = 10011 \sim \delta^{13} \Rightarrow [L_{13} \to \delta^{13}] \\ L_5 \to \delta^5 \sim 11011 \Rightarrow \emptyset.$$
- $$E_3 = 00100 \sim \delta^{28} \Rightarrow [L_{28} \to \delta^{28}] \\ L_{16} \to \delta^{16} \sim 10000 \\ \Rightarrow \begin{cases} L_4 \to \delta^4 \sim 11100 \Rightarrow \begin{cases} L_3 \to \delta^3 \sim 11101 \Rightarrow \emptyset \\ L_{19} \to \delta^{19} \sim 01101 \Rightarrow \emptyset \\ L_{23} \to \delta^{23} \sim 01001 \Rightarrow \emptyset \end{cases} \\ L_{24} \to \delta^{24} \sim 01000 \Rightarrow L_{15} \to \delta^{15} \sim 10001 \Rightarrow \emptyset. \end{cases}$$

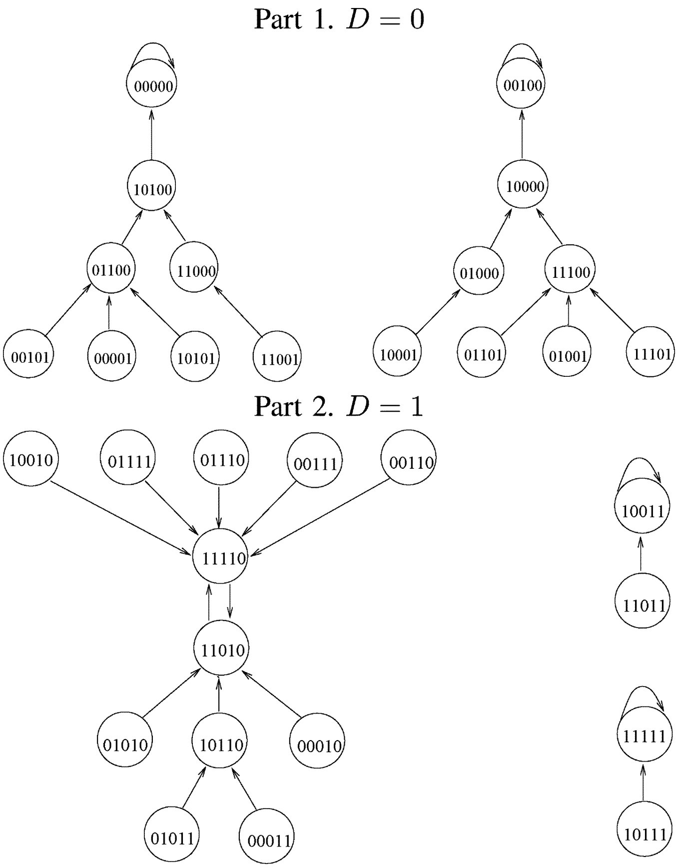

Fig. 5. State-transition diagram.

- $$E_4 = 00000 \sim \delta^{32} \Rightarrow [L_{32} \rightarrow \delta^{32}]$$
$$L_{12} \rightarrow \delta^{12} \sim 10100$$
$$\Rightarrow \begin{cases} L_{20} \rightarrow \delta^{20} \sim 01100 \Rightarrow \begin{cases} L_{11} \rightarrow \delta^{11} \sim 10101 \Rightarrow \emptyset \\ L_{27} \rightarrow \delta^{27} \sim 00101 \Rightarrow \emptyset \\ L_{31} \rightarrow \delta^{31} \sim 00001 \Rightarrow \emptyset \end{cases} \\ L_8 \rightarrow \delta^8 \sim 11000 \Rightarrow L_7 \rightarrow \delta^7 \sim 11001 \Rightarrow \emptyset. \end{cases}$$
- Next, we consider two points on a cycle: $C_1 = 11010$ and $C_2 = 11110$. For C_1
$$C_1 = 11010 \sim \delta^6$$
$$\Rightarrow \begin{cases} [L_2 \rightarrow \delta^2 \sim 11110] \\ L_{10} \rightarrow \delta^{10} \sim 10110 \Rightarrow \begin{cases} L_{21} \rightarrow \delta^{21} \sim 01011 \Rightarrow \emptyset \\ L_{29} \rightarrow \delta^{29} \sim 00011 \Rightarrow \emptyset \end{cases} \\ L_{22} \rightarrow \delta^{22} \sim 01010 \Rightarrow \emptyset \\ L_{31} \rightarrow \delta^{31} \sim 00010 \Rightarrow \emptyset. \end{cases}$$
- $$C_2 = 11110 \sim \delta^2 \Rightarrow \begin{cases} [L_6 \rightarrow \delta^6 \sim 11010] \\ L_{14} \rightarrow \delta^{14} \sim 11010 \Rightarrow \emptyset \\ L_{17} \rightarrow \delta^{17} \sim 01111 \Rightarrow \emptyset \\ L_{18} \rightarrow \delta^{18} \sim 01110 \Rightarrow \emptyset \\ L_{25} \rightarrow \delta^{25} \sim 00111 \Rightarrow \emptyset \\ L_{26} \rightarrow \delta^{26} \sim 00110 \Rightarrow \emptyset. \end{cases}$$

The following state transition diagram from [14] verifies our conclusion.

What is significant in this example is the following observation. There is a smallest "cycle": fixed point D. From Fig. 5, one sees easily that for $D = 0$ and $D = 1$ the topological structures of the state–space graphs are completely different.

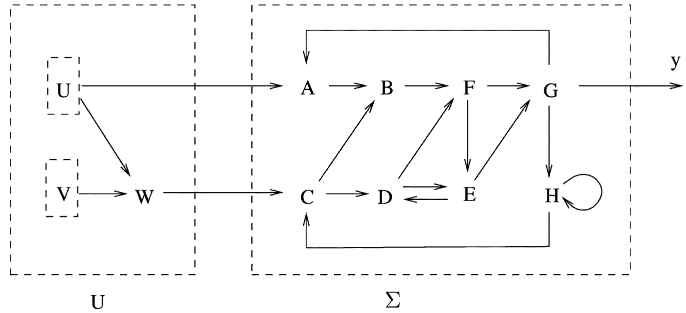

Fig. 6. Gene and protein signaling activity patterns.

Next, we analyze a system, which is used to simulate gene and protein signaling activity patterns [9].

Example 5.2: The network depicted in Fig. 6 and Table II is presented in [9] to simulate gene and protein signaling activity patterns within a small model Boolean network. For notational brevity, we use A for "Erk," B for "cyclin D1," C for "p27," D for "cyclin E," E for "E2F," F for "pRb," G for "S phase genes," U for "growth factors," V for cell "shape(spreading)," and W for "X." We refer to [9] for the biological meanings of the notations.

Then, the logical equation is expressed as
$$\begin{cases} A(t+1) = \neg(U(t) \rightarrow G(t)) \\ B(t+1) = \neg(A(t) \rightarrow C(t)) \\ C(t+1) = H(t) \rightarrow W(t) \\ D(t+1) = E(t) \rightarrow C(t) \\ E(t+1) = \neg(D(t) \rightarrow F(t)) \\ F(t+1) = \neg(B(t) \wedge D(t)) \\ G(t+1) = \neg(F(t) \rightarrow E(t)) \\ H(t+1) = \neg(H(t) \rightarrow G(t)). \end{cases} \tag{38}$$

As for the control network, we have
$$\begin{cases} U(t+1) = g_1(U(t)) \\ V(t+1) = g_2(V(t)) \\ W(t+1) = g_3(U(t), V(t)). \end{cases} \tag{39}$$

In a matrix form, we have an algebraic equation as
$$\begin{cases} A(t+1) = M_n M_i U(t) G(t) \\ B(t+1) = M_n M_i A(t) C(t) \\ C(t+1) = M_i H(t) W(t) \\ D(t+1) = M_i E(t) C(t) \\ E(t+1) = M_n M_i D(t) F(t) \\ F(t+1) = M_n M_c B(t) D(t) \\ G(t+1) = M_n M_i F(t) E(t) \\ H(t+1) = M_n M_i H(t) G(t). \end{cases} \tag{40}$$

As in [9], we first set the control network as
$$\begin{cases} U(t+1) = U(t) \\ V(t+1) = V(t) \\ W(t+1) = U(t) \wedge V(t). \end{cases} \tag{41}$$

Case 1: $U(0) = V(0) = [1, 0]^T$, $\sigma_1 = \sigma_2 =$ identity, i.e., $U(t)$ and $V(t)$ equal to $[1, 0]^T$ constantly.

TABLE II
LOGICAL RELATIONS

network element	W	A	B	C	D	E	F	G	H
input 1	U	U	A	H	E	D	B	F	H
input 2	V	G	C	W	C	F	D	E	G
boolean function	and	not if	not if	implicate	implicate	not if	nand	not if	not if

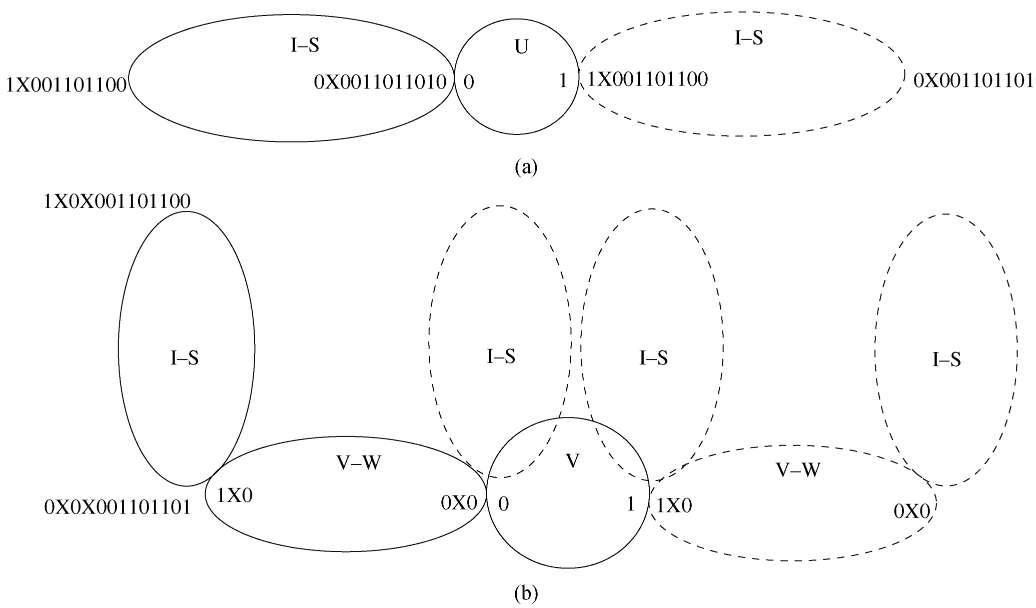

Fig. 7. Chaired circles. (a) Circles in $U \subset I - S$ spaces. (b) Circles in $V \subset V \times W \subset I - S$ spaces.

Plugging them into (39) yields the system

$$L(U(t), W(t)) = L\left(\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}\right).$$

In calculation, a control can be treated as a logical operator, so the routine for calculating the network transition matrix remains available. Then, it is easy to get the following result:

- the only attractor is a fixed point 00110110;
- $L^{10} = L^{11}$ and the transient period is $T_t = 10$.

Case 2: $U(0) = [1, 0]^T$ and $V(0) = [0, 1]$. In this case, we have the same conclusion.

Case 3: $U(0) = [0, 1]^T$, then we always have $W(t) = [0, 1]^T$, $t \geq 1$. The conclusion is as follows:

- the only attractor is a fixed point 00110110;
- $L^6 = L^7$ and the transient period is $T_t = 6$. (Taking $W(0)$ into consideration, T_t should be 7.)

Next, we assume that the control network is

$$\begin{cases} U(t+1) = \neg U(t) \\ V(t+1) = \neg V(t) \\ W(t+1) = V(t) \wedge W(t). \end{cases} \tag{42}$$

Then, we have two sequences of nested invariant subspaces. We consider them separately. Consider the first chair, which is

$$\begin{aligned} V_1 &= \mathrm{Span}\{U\} \\ \subset V_2 &= \mathrm{Span}\{A, B, C, D, E, F, G, H, U, V, W\}. \end{aligned}$$

In V_1, we have an obvious cycle: $0 \to 1 \to 0$. For $U = 0$, a routine computation shows that there is only a cycle of length 2, which is

$$0011011010 \to 0011011000 \to 0011011010.$$

$L(0)$ is a 1024×1024 matrix. We omit it here. But we can calculate that $L(0)^7 = L(0)^9$ and $T_t = 7$. Similarly, for $U = 1$, we have the same cycle. $L(1)^{11} = L(1)^{13}$ and $T_t = 11$.

Finally, let $\Psi = L(1)L(0)$. Then, Ψ has only one fixed point 0011011010. We conclude that overall in U space we have only one cycle $0 \to 1 \to 0$; and in the whole space, we have only one product cycle

$$0 \times 0011011010 \to 1 \times 0011011000 \to 0 \times 0011011010.$$

They are depicted in Fig. 7(a), where $I - S$ is the overall input-state space.

Next, we consider the second chair, which is

$$\begin{aligned} V_1 &= \mathrm{Span}\{V\} \subset V_2 = \mathrm{Span}\{V, W\} \\ \subset V_3 &= \mathrm{Span}\{A, B, C, D, E, F, G, H, U, V, W\}. \end{aligned}$$

First, there is a trivial cycle in V space as: $0 \to 1 \to 0$. Then, in $V \times W$ space, it is easy to calculate that

$$L(0) = \begin{bmatrix}0 & 0\\1 & 1\end{bmatrix} \quad L(1) = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}.$$

So

$$\Psi = L(1)L(0) = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$$

which has a unique fixed point $(0, 1) \sim 0$. We conclude that in $V \times W$ space we have only one cycle: $0 \times 0 \to 1 \times 0 \to 0 \times 0$. Finally, we consider the space $V \times W \times ABCDEFGHU$. Calculating $\Psi = L(0 \times 0)L(1 \times 0)$, it is easy to find that the only cycle is a fixed point: 001101101. We conclude that there is only one cycle of length 2 in the overall product space, which is $0 \times 0 \times 001101101 \to 1 \times 0 \times 001101100$. Circles in nested subspaces are depicted in Fig. 7(b).

VI. Conclusion

In this paper, we first reviewed the main results of [5]: how to convert a logical form of Boolean equation into an algebraic form. Then, we proposed a framework for control Boolean networks. The structure of the cycles in the input-state space was obtained. Using the input-state technique, we investigated the general structure of Boolean networks. Then, as the network has a cascade structure, a structure of chaired cycles, called *rolling gears*, has been revealed. It was shown that the rolling gears structure is the explanation that tiny attractors decide the order of networks.[1]

References

[1] D. W. Barnes, *An Algebraic Introduction to Mathematical Logic*. New York: Springer-Verlag, 1975.

[2] A. Beg, P. W. C. Prasad, and A. Beg, "Applicability of feed-forward and recurrent neural networks to Boolean function complexity modeling," *Expt. Syst. Appl.*, vol. 34, no. 4, pp. 2436–2443, 2008.

[3] D. Cheng, "On logic-based intelligent systems," in *Proc. Int. Conf. Component Anal.*, Budapest, Hungary, 2005, pp. 71–76.

[1]STP Toolbox for the related computations is available at http://lsc.amss.ac.cn/~dcheng/

[4] D. Cheng and H. Qi, "Matrix expression of logic and fuzzy control," in *Proc. 44th IEEE Conf. Decision Control*, Seville, Italy, 2005, pp. 3273–3278.

[5] D. Cheng and H. Qi, "Linear representation of dynamics of Boolean networks," *IEEE Trans. Autom. Control*, submitted for publication.

[6] C. Farrow, J. Heidel, H. Maloney, and J. Rogers, "Scalar equations for synchronous Boolean networks with biological applications," *IEEE Trans. Neural Netw.*, vol. 15, no. 2, pp. 348–354, Mar. 2004.

[7] A. G. Hamilton, *Logic for Mathematicians*, revised ed. Cambridge, U.K.: Cambridge Univ. Press, 1988.

[8] C. Y.-F Ho, B. W.-K. Ling, H.-K. Lam, and M. H. U. Nasir, "Global convergence and limit cycle behavior of weights of perceptron," *IEEE Trans. Neural Netw.*, vol. 19, no. 6, pp. 938–947, Jun. 2008.

[9] S. Huang and I. Ingber, "Shape-dependent control of cell growth, differentiation, and apotosis: Switching between attractors in cell regulatory networks," *Exp. Cell Res.*, vol. 261, pp. 91–103, 2000.

[10] J. Heidel, J. Maloney, J. Farrow, and J. Rogers, "Finding cycles in synchronous Boolean networks with applications to biochemical systems," *Int. J. Bifurcat. Chaos*, vol. 13, no. 3, pp. 535–552, 2003.

[11] T. Ideker, T. Galitski, and L. Hood, "A new approach to decoding life: Systems biology," *Annu. Rev. Genomics Human Gen.*, vol. 2, pp. 343–372, 2001.

[12] S. A. Kauffman, *At Home in the Universe*. Oxford, U.K.: Oxford Univ. Press, 1995.

[13] J. Lian, Y. Lee, S. D. Sudhoff, and S. H. Zak, "Self-organizing radial basis function for real-time approximation of continuous-time dynamical systems," *IEEE Trans. Neural Netw.*, vol. 19, no. 3, pp. 460–474, Mar. 2008.

[14] E. Paszek, Boolean Networks, 2008 [Online]. Available: http://cnx.org/content/m12394/latest/

[15] M. M. Waldrop, *Complexity*. New York: Touchstone, 1992.

[16] S. Wu and T. W. S. Chow, "Self-organizing and self-evolving neurons: A new neural netwrok for optimization," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 385–396, Mar. 2008.

Daizhan Cheng (M'01–SM'01–F'06) received the Ph.D. degree in systems science from Washington University, St. Louis, in 1985.

Currently, he is a Professor with Institute of Systems Science, Chinese Academy of Sciences, Beijing, China. His research interests include nonlinear systems, numerical method, systems biology, etc.

Dr. Cheng is Chairman of the Technical Committee on Control Theory (since 2003), Chinese Association of Automation, and the International Federation of Automatic Control (IFAC) Fellow.