Automatica 59 (2015) 54–59

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Output tracking control of Boolean control networks via state feedback: Constant reference signal case[☆]

Haitao Li[a,b], Yuzhen Wang[b,1], Lihua Xie[c]

[a] School of Mathematical Science, Shandong Normal University, Jinan 250014, PR China
[b] School of Control Science and Engineering, Shandong University, Jinan 250061, PR China
[c] EXQUISITUS, Center for E-City, School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore

ARTICLE INFO

Article history:
Received 28 March 2014
Received in revised form
15 May 2015
Accepted 21 May 2015
Available online 17 June 2015

Keywords:
Boolean control network
Output tracking control
State feedback
Semi-tensor product of matrices

ABSTRACT

This paper investigates the state feedback based output tracking control of Boolean control networks (BCNs) with a constant reference signal by using the semi-tensor product method. Based on the algebraic expression of BCNs and by constructing a series of reachable sets, a general procedure is proposed for the design of the state feedback laws for BCNs to track a constant reference signal. The study of an illustrative example shows that the obtained new results are effective in designing state feedback based output tracking controllers for BCNs to track a constant reference signal.

1. Introduction

Since Kauffman's pioneering work (Kauffman, 1969) in the Boolean model of gene regulatory networks, the study of Boolean networks has attracted great attention of biologists, physicists and systems scientists. Consequently, many excellent results have been established for Boolean networks (Akutsu, Hayashida, Ching, & Ng, 2007; Ay, Xu, & Kahveci, 2009; Chaves, 2009; Drossel, Mihaljev, & Greil, 2005; Xiao, 2009; Xiao & Dougherty, 2007). In a Boolean network, each gene can take two possible values, 1 and 0, and its value (1 or 0) indicates its measured abundance (expressed or unexpressed; high or low). From a graphical perspective, genes in a Boolean network are nodes in this network and edges describe regulatory relationships between genes.

As is well known, the ultimate goal of modeling gene regulatory networks as Boolean networks is to design effective therapeutic intervention strategies to influence the network dynamics to avoid undesirable cellular states. Hence, in order to manipulate Boolean networks, binary control inputs and outputs are added to the network dynamics, which yields Boolean control networks (BCNs). The control of BCNs is a fundamental issue in both systems biology and control theory. However, due to the lack of effective tools to deal with logical dynamics, the control of BCNs has been a challenging problem for a long time until the introduction of the semi-tensor product method (Cheng, Qi, & Li, 2011). The feature of this method is that one can convert the dynamics of a Boolean (control) network into a linear (bilinear) discrete-time system. Then, one can analyze Boolean (control) networks by using the classical control theory. Up to now, there have been many interesting works on the control of BCNs via this novel method, which include controllability and observability (Chen & Sun, 2014; Cheng, Li, & Qi, 2010; Cheng & Qi, 2009, 2010; Feng, Yao, & Cui, 2012; Fornasini & Valcher, 2013a; Laschov & Margaliot, 2012; Li & Sun, 2011a,b; Li & Wang, 2012; Zhang & Zhang, 2013; Zhao, Cheng, & Qi, 2010), disturbance decoupling (Cheng, 2011; Yang, Li, & Chu, 2013), optimal control (Fornasini & Valcher, 2014; Laschov & Margaliot, 2011; Zhao, Li, & Cheng, 2011), stability and stabilization (Cheng, Qi, Li, & Liu, 2011; Li & Wang, 2013; Li, Yang, & Chu, 2013), and other control problems (Cheng & Xu, 2013; Cheng & Zhao, 2011; Fornasini & Valcher, 2013b; Li & Chu, 2012; Wang, Zhang, & Liu, 2012; Xu & Hong, 2013; Zhang, 2012; Zhang & Feng, 2013; Zhao, Kim, & Filippone, 2013; Zou & Zhu, 2014).

It is noted that in many practical gene regulatory networks, the state variables cannot be measured directly due to the limitation

☆ The research was supported by the National Research Foundation of Singapore under grant NRF-CRP8-2011-03, the National Natural Science Foundation of China under grants 61374065 and 61320106011, and the Research Fund for the Taishan Scholar Project of Shandong Province. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Yoshito Ohta under the direction of Editor Richard Middleton.

E-mail addresses: haitaoli09@gmail.com (H. Li), yzwang@sdu.edu.cn (Y. Wang), ELHXIE@ntu.edu.sg (L. Xie).

1 Tel.: +86 531 88392515; fax: +86 531 88392205.

http://dx.doi.org/10.1016/j.automatica.2015.06.004
0005-1098/

of measurement conditions and the impact of immeasurable variables. In this case, one can use the measured outputs to track a desirable reference signal which corresponds to some desirable states. Thus, it is meaningful for us to design suitable controllers (therapeutic intervention) that steer the output of BCNs to an expected reference signal, called the output tracking control problem of BCNs in this paper. It should be pointed out that several types of therapeutic intervention have been proposed for Boolean networks until now. Among them, the theory of automatic control, such as the controllability and the optimal control, is an important type of therapeutic intervention, and thereby many suitable control strategies have been obtained based on the classical control theory (Akutsu et al., 2007; Choudhary, Datta, Bittner, & Dougherty, 2006). As one of the most important issues in the control theory, it is believed that the output tracking control can provide an effective way for the design of therapeutic intervention. However, to our best knowledge, there are no results available on the output tracking control of BCNs.

In this paper, using the semi-tensor product method, we investigate how to design output tracking controllers for BCNs to track a constant reference signal. We propose a general procedure to design state feedback based output tracking controllers for BCNs. The key idea of this procedure is to stabilize the BCN to a set of states whose outputs are the given constant reference signal. Although this procedure is a generalization of the state feedback stabilization control design method proposed in Fornasini and Valcher (2013b) and Li et al. (2013), it has the following two differences/novelties:

(1) The procedure proposed in this paper has wider applications (such as the output tracking control, and the stabilization to a stable region) because it can stabilize the BCN to a set of states. When the set of states only contains an element, the procedure degenerates to that of Fornasini and Valcher (2013b) and Li et al. (2013).
(2) In our procedure, the set of states that is stabilized to is not fixed, while the equilibrium in Fornasini and Valcher (2013b) and Li et al. (2013) is fixed. How to determine a proper set of states whose outputs are the given constant reference signal is a very challenging problem in our procedure. In this paper, we solve this problem by using the input-state incidence matrix introduced in Zhao et al. (2010) (please see Theorem 2).

The rest of this paper is organized as follows. Section 2 formulates the output tracking control problem studied in this work. Section 3 investigates the output tracking control of BCNs via state feedback and presents the main results of this paper. An illustrative example is given to support our new results in Section 4, which is followed by a brief conclusion in Section 5.

Notation: The notation of this paper is fairly standard. $\mathcal{D} := \{1, 0\}$. $\Delta_n := \{\delta_n^k : k = 1, \dots, n\}$, where δ_n^k denotes the kth column of the identity matrix I_n. For compactness, $\Delta := \Delta_2$. An $n \times t$ matrix M is called a logical matrix, if $M = [\delta_n^{i_1}\ \delta_n^{i_2}\ \cdots\ \delta_n^{i_t}]$, and we express M briefly as $M = \delta_n[i_1\ i_2\ \cdots\ i_t]$, and denote the set of $n \times t$ logical matrices by $\mathcal{L}_{n\times t}$. $Col_i(A)$ denotes the ith column of the matrix A, and $Row_i(A)$ stands for the ith row of the matrix A. $\mathbf{0}_n := \big[\underbrace{0\ 0\ \cdots\ 0}_{n}\big]$. "$\neg$", "$\wedge$" and "$\vee$" denote Negation, Conjunction and Disjunction, respectively.

2. Problem formulation

Consider the following Boolean control network:

$$\begin{cases} x_1(t+1) = f_1(X(t), U(t)), \\ x_2(t+1) = f_2(X(t), U(t)), \\ \ \vdots \\ x_n(t+1) = f_n(X(t), U(t)); \\ y_j(t) = h_j(X(t)), \quad j = 1, \dots, p, \end{cases} \tag{1}$$

where $X(t) = (x_1(t), x_2(t), \dots, x_n(t)) \in \mathcal{D}^n$, $U(t) = (u_1(t), \dots, u_m(t)) \in \mathcal{D}^m$ and $Y(t) = (y_1(t), \dots, y_p(t)) \in \mathcal{D}^p$ are the state, the control input and the output of the system (1), respectively, and $f_i : \mathcal{D}^{m+n} \mapsto \mathcal{D}$, $i = 1, \dots, n$ and $h_j : \mathcal{D}^n \mapsto \mathcal{D}$, $j = 1, \dots, p$ are logical functions. Given a control sequence $\{U(t) : t \in \mathbb{N}\}$, denote the state trajectory of the system (1) starting from an initial state $X(0) \in \mathcal{D}^n$ by $X(t; X(0), U)$, and the output trajectory of the system (1) starting from $X(0) \in \mathcal{D}^n$ by $Y(t; X(0), U)$.

The output tracking control problem studied in this paper is to design a state feedback control in the form of

$$\begin{cases} u_1(t) = k_1(X(t)), \\ \ \vdots \\ u_m(t) = k_m(X(t)), \end{cases} \tag{2}$$

such that the output of the closed-loop system consisting of the system (1) and the control (2) tracks a given constant reference signal $Y_r = (y_1^r, \dots, y_p^r) \in \mathcal{D}^p$, that is, there exists an integer $\tau > 0$ such that $Y(t; X(0), U) = Y_r$ holds for $\forall\, X(0) \in \mathcal{D}^n$ and $\forall\, t \ge \tau$, where $k_i : \mathcal{D}^n \mapsto \mathcal{D}$, $i = 1, \dots, m$ are logical functions to be determined.

In the following, we convert the system (1) and the state feedback control (2) into equivalent algebraic forms, respectively. To this end, we first recall the definition and some properties of the semi-tensor product of matrices.

Definition 1 (*Cheng et al., 2011*). The semi-tensor product of two matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$ is

$$A \ltimes B = (A \otimes I_{\frac{\alpha}{n}})(B \otimes I_{\frac{\alpha}{p}}), \tag{3}$$

where $\alpha = lcm(n, p)$ is the least common multiple of n and p, and $\otimes$ is the Kronecker product.

It is noted that when $n = p$, the semi-tensor product of A and B becomes the conventional matrix product. Thus, the semi-tensor product is a generalization of the conventional matrix product. We can simply call it "product" and omit the symbol "$\ltimes$" if no confusion arises in the following.

Proposition 1 (*Cheng et al., 2011*). *The semi-tensor product of matrices has the following properties:*

(i) *Let $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{p\times q}$ and $C \in \mathbb{R}^{r\times s}$. Then $(A \ltimes B) \ltimes C = A \ltimes (B \ltimes C)$.*
(ii) *Let $X \in \mathbb{R}^{t\times 1}$ be a column vector and $A \in \mathbb{R}^{m\times n}$. Then*

$$X \ltimes A = (I_t \otimes A) \ltimes X. \tag{4}$$

By identifying $1 \sim \delta_2^1$ and $0 \sim \delta_2^2$, we have $\Delta \sim \mathcal{D}$, where "$\sim$" denotes two different expressions of the same thing. In most places of this work, we use δ_2^1 and δ_2^2 to express logical variables and call them the vector form of logical variables. The following lemma is fundamental for the algebraic expression of logical functions.

Lemma 1 (*Cheng et al., 2011*). *Let $f(x_1, x_2, \dots, x_s) : \mathcal{D}^s \mapsto \mathcal{D}$ be a logical function. Then, there exists a unique matrix $M_f \in \mathcal{L}_{2\times 2^s}$, called the structural matrix of f, such that*

$$f(x_1, x_2, \dots, x_s) = M_f \ltimes_{i=1}^{s} x_i, \tag{5}$$

where $x_i \in \Delta$ and $\ltimes_{i=1}^{s} x_i = x_1 \ltimes \cdots \ltimes x_s$.

Using the vector form of logical variables and setting $x(t) = \ltimes_{i=1}^{n} x_i(t) \in \Delta_{2^n}$, $u(t) = \ltimes_{i=1}^{m} u_i(t) \in \Delta_{2^m}$ and $y(t) = \ltimes_{i=1}^{p} y_i(t) \in \Delta_{2^p}$, by Lemma 1, one can convert (1) and (2) into

$$\begin{cases} x(t+1) = Lu(t)x(t), \\ y(t) = Hx(t), \end{cases} \tag{6}$$