Self-Triggered Scheduling for Boolean Control Networks

Min Meng, Gaoxi Xiao, Senior Member, IEEE and Daizhan Cheng, Fellow, IEEE

Abstract—It has been shown that self-triggered control has the ability to reduce computational loads and deal with the cases with constrained resources by properly setting up the rules for updating the system control when necessary. In this paper, self-triggered stabilization of Boolean control networks (BCNs), including deterministic BCNs, probabilistic BCNs and Markovian switching BCNs, is first investigated via semi-tensor product of matrices and Lyapunov theory of Boolean networks. The self-triggered mechanism with the aim to determine when the controller should be updated is given based on the decrease of the corresponding Lyapunov functions between two successive sampling times. We show that the self-triggered controllers can be chosen as the conventional controllers without sampling, and also can be optimally constructed based on the triggering conditions.

Index Terms—Boolean control networks, self-triggered scheduling, semi-tensor product, Lyapunov function.

I. INTRODUCTION

Boolean networks have attracted considerable attention due to their wide applications in various fields such as gene regulatory networks [1], smart home [2] and game theory [3]–[5], etc. Extensive studies have been conducted on analysis and control problems of Boolean networks by semi-tensor product of matrices [6], [7] in the last decade, with different focuses on system stability, optimization, observability, controllability and so on. Readers may refer to [8]–[17] and the references therein for more details.

In most existing references on stabilization and controller design for Boolean control networks (BCNs), it is required that the states at all the discrete time should be accessible. In practice, however, the measurable information may be scarce, e.g., caused by constrained resources such as a limited lifetime of battery-powered devices; consequently, the control may not be implemented every time to guarantee the desired closed-loop performance. For such applications, it is necessary to develop control techniques depending on the measurable states being available at partial discrete time.

Periodic sampling, which is a special case where the measurements are available periodically, has been applied to study state feedback stabilization for BCNs [18], [19]. The work was then extended to non-periodical sampling [20], which is also prescheduled. Such sampling intervals can be regarded as exogenous signals which are deterministic regardless of whether the systems need attention. On the other hand, however, the sampling time is always unknown in advance in event-based cases, where the next sampling time at which the control is updated always hinges on the control itself and a state-dependent criterion in a way that the stability of the closed-loop system is not destroyed [21]–[23]. Related works on event-based control of Boolean networks can be found in [24]–[28]. In [24], the disturbance decoupling problem was studied by event-triggered control and the triggering condition as a rank condition of the network transition matrices. In [25], the authors designed the triggering times based on the Hausdorff distance to study robust control of BCNs with disturbances. Subsequently, Zhu and Lin in [26] obtained an optimal event-triggered control strategy for stabilization of BCNs by constructing the weighted digraph and the hypergraph for the BCN and applying the shortest path algorithm to the hypergraph. The idea of event-triggered control was also extended to study synchronization of drive-response BCNs [27] and robust invariance of probabilistic BCNs [28]. Such results can indeed reduce the number of samples while still fulfilling the requests. The event-triggered control, with all its advantages, has to depend on constant measurements to detect whether the triggering conditions are fulfilled. To avoid constant measurements, self-triggered sampling scheduling was proposed [29], [30] with the advantage that the next sampling time $t_{k+1}$ can be determined in advance only based on the state and controller at the current sampling time $t_k$. To our best knowledge, there are no references on self-triggered control for BCNs, which motives our study in this paper for improving the existing periodic/event-triggered sampling schemes for BCNs.

As is well known, in practical systems, there may be several evolutiveational strategies for one state variable at every discrete time. For example, the bacteriophage $\lambda$ in genetic regulatory networks possesses different behaviors (lysis and lysogeny). Since a series of molecular processes in genetic regulatory networks is always affected by some intrinsic fluctuations and extrinsic perturbations with stochastic factors, probabilistic/Markovian switching Boolean networks may have advantages in modeling the rule-based properties and the uncertainties. Stability and stabilization for probabilistic/Markovian switching Boolean networks have been investigated in [31]–[34], while the self-triggered control has not been considered.

In this paper, for the first time to the best of our knowledge, we investigate self-triggered control for BCNs based on Lyapunov functions for Boolean networks. Three kinds of BCNs, namely deterministic BCNs, probabilistic BCNs and Markovian switching BCNs, respectively, are considered. Lyapunov functions for deterministic and Markovian switching Boolean networks were, respectively, proposed in [35] and [36]. However, there is no systematic analysis on Lyapunov stability for all the different classes of Boolean networks. In this paper, we introduce and conclude Lyapunov stability theory for Boolean networks, where the Lyapunov function for probabilistic Boolean networks is given for the first time. Then the self-triggered conditions are designed hinges on the decrease of the corresponding Lyapunov functions between two successive samplings. We also prove that the given stabilization controllers, together with self-triggered updated scheduling, can ensure that the control strategies are well defined and the resulted closed-loop systems are stable. To construct optimal self-triggered controllers, the co-design of self-triggered sampling scheme and the related controllers is also considered.

In summary, the main contributions of this paper are twofold: i) We present Lyapunov stability theory for probabilistic Boolean networks in the form of linear inequalities, which can be easily applied to constructed Lyapunov function for probabilistic Boolean networks. ii) We propose self-triggered scheduling for BCNs, based on which the limited resources case can be easily solved and the next triggering time can be pre-determined. Our method outperforms the sampled-data and event-triggered methods.
The remainder of this paper is organized as follows. Section II introduces some preliminary results about semi-tensor product of matrices. In Section III, we introduce the Lyapunov stability theory for three kinds of Boolean networks. In Section IV, self-triggered scheduling and control design are analyzed for BCNs, probabilistic BCNs and Markovian switching BCNs, respectively. Finally, a brief conclusion is given in Section V.

Notations. \( \mathbb{R}^n \) and \( \mathbb{R}^{m \times n} \) denote the sets of n-dimensional column vectors and \( m \times n \) real matrices, respectively. Denote \( \mathbb{B} = \{0, 1\} \). The symbol \( \mathbb{B}^{m \times n} \) represents the set of \( n \times m \) matrices with every element being in \( \mathbb{B} \). The matrices in \( \mathbb{B}^{n \times m} \) are called Boolean matrices, \( \mathbb{B}^n := \mathbb{B}^{n \times 1} \). The \( i \)th column of the identity matrix \( I_n \) is defined as \( \delta_i^n \), \( i = 1, 2, \ldots, n \). Denote \( \Delta_n := \{\delta_i^n \mid i = 1, 2, \ldots, n\} \). A matrix \( L \in \mathbb{B}^{n \times r} \), written as \( L = [\delta^n_1, \delta^n_2, \ldots, \delta^n_r] \), is called a logical matrix and can be rewritten as \( L = \delta^n_{[i_1, i_2, \ldots, i_r]} \) for simpler notation. Denote by \( \mathcal{L}_{n \times r} \) the set of \( n \times r \) logical matrices. Col(\( L \)) represents the \( i \)th column of \( L \) and Col(\( L \)) is the set of columns of \( L \). When \( \phi \) is a matrix of \( \mathbb{B}^n \), \( \mathcal{M} \) is called a logical product of matrices, used in this paper.

**Definition 1** [56, 77]: The semi-tensor product of matrices \( M \in \mathbb{R}^{a \times b} \) and \( N \in \mathbb{R}^{c \times d} \), denoted by \( M \times N \), is defined as

\[
M \times N = (M \otimes I_{b/c}) (N \otimes I_{a/b}),
\]

where \( b \) is the least common multiple of \( b \) and \( c \).

The semi-tensor product of matrices in Definition 1 generalizes the traditional matrix product \( M \times N = MN \) when \( b = c \). Therefore, most of the matrix products appearing in this paper can be regarded as semi-tensor product and the symbol \( \times \) is omitted if no confusion arises. Further discussions on properties and applications of semi-tensor product can be referred to [56, 77].

The essential step of using semi-tensor product of matrices to study Boolean networks is to define a bijection mapping from \( \mathbb{B}^n \) to \( \Delta_2^n \), i.e., \( 0 \sim \delta_2^n \sim 1 \). Then we can get a bijection from \( \mathbb{B}^n \) to \( \Delta_2^n \), denoted by \( \phi_n : \mathbb{B}^n \rightarrow \Delta_2^n \), which is defined as

\[
\phi_n(X) = \left( X_1 \ X_2 \right) \times \left( X_2 \ X_3 \right) \times \cdots \times \left( X_n \ \bar{X}_n \right) \in \Delta_2^n,
\]

where \( X = (X_1, X_2, \ldots, X_n)^T \in \mathbb{B}^n \) and \( \bar{X}_i = 1 - X_i, i = 1, 2, \ldots, n \). Note that a Boolean function with \( n \) variables is a mapping from \( \mathbb{B}^n \) to \( \mathbb{B} \). One important lemma for equivalently converting the original logical form of Boolean networks to an algebraic expression is presented as follows.

**Lemma 1** [56, 77]: For a Boolean function \( \psi : \mathbb{B}^n \rightarrow \mathbb{B} \), there exists a unique matrix \( M_\psi \in \mathcal{L}_{2^n \times 2^n} \), which is named as the structure matrix of \( \psi \), such that

\[
\phi_1(\psi(X)) = M_\psi \phi_n(X),
\]

where \( \phi_1, \phi_n \) are defined in [1].

**III. Lyapunov Stability Theory**

This section will introduce and give the Lyapunov stability theory for three classes of Boolean networks. The Lyapunov function for probabilistic Boolean networks is defined for the first time, while those for the other two kinds of Boolean networks can be found in [35, 37, 38].

A. Lyapunov function for Boolean networks

A Boolean network with \( n \) nodes is given as

\[
x(t + 1) = f(x(t)),
\]

where \( x(t) \in \mathbb{B}^n \) and \( f : \mathbb{B}^n \rightarrow \mathbb{B}^n \) is a Boolean vector function. Based on the semi-tensor product in Definition 1 and Lemma 1, the algebraic form of Boolean network (1) can be equivalently rewritten as

\[
x(t + 1) = F x(t),
\]

where \( x(t) = \phi_n (X(t)) \in \Delta_2^n \), and \( F \) is in \( \mathcal{L}_{2^n \times 2^n} \), called the transition matrix of (4). Split \( F \) as

\[
F = \begin{bmatrix}
F_{11} & F_{12} \\
F_{21} & F_{22}
\end{bmatrix},
\]

where \( F_{11} \in \mathbb{B}^{(2^n-1) \times (2^n-1)} \) and \( F_{22} \in \mathbb{B} \). Then one can verify that one necessary condition for stability at \( \delta_2^n \) of Boolean network (4) is that \( \delta_2^n \) is a fixed point of (4), which is equivalent to \( F \delta_2^n = \delta_2^n \), i.e., \( F_{11} = 0_{2^n-1} \) and \( F_{22} = 1 \). Reviewing the Lyapunov theory proposed in [35, 37, 38], Boolean network (4) is stable at the point \( \delta_2^n \) if and only if there exists a Lyapunov function of Boolean network (4). \( V_1(x(t)) \), which is defined to satisfy

- \( V_1(x(t)) > 0 \) for \( x(t) \neq \delta_2^n \) and \( V_1(x(t)) = 0 \) for \( x(t) = \delta_2^n \);
- \( \Delta V_1(x(t)) < 0 \) for \( x(t) \neq \delta_2^n \) and \( \Delta V_1(x(t)) = 0 \) for \( x(t) = \delta_2^n \), where \( \Delta V_1(x(t)) := V_1(x(t + 1)) - V_1(x(t)) \).

Following these criteria, one Lyapunov function can be constructed as

\[
V_1(x(t)) = \lambda^T x(t),
\]

where \( \lambda = (\lambda_1, 0)^T \in \mathbb{R}^{n} \) satisfies

\[
\lambda_1 > 0,
\]

\[
F_{11} \lambda_1 - \lambda_1 < 0.
\]

B. Lyapunov function for probabilistic Boolean networks

If the update strategy of a Boolean network is not deterministic and belongs to a set of possible update strategies with certain probability distribution, then the Boolean network becomes a probabilistic Boolean network. Consider a probabilistic Boolean network with \( n \) nodes and \( s \) possible update strategies as

\[
Y(t + 1) = g(y(Y(t))),
\]

where \( Y(t) \in \mathbb{B}^n \), and \( g(t) \in \{g_1, g_2, \ldots, g_s\} \) with \( g_i : \mathbb{B}^n \rightarrow \mathbb{B}^n \) being a Boolean vector function, \( i = 1, 2, \ldots, s \). Moreover, for every time \( t \), \( \Pr\{g(t) = g_i\} = p_i \), where \( p_i \geq 0 \) and \( \sum_{i=1}^s p_i = 1 \). Without loss of the generality, it is assumed that \( p_i > 0 \) for every \( i = 1, 2, \ldots, s \) since if \( p_i = 0 \), then we can consider the possible update strategy set as \( \{g_1, g_2, \ldots, g_s\} \setminus \{g_i\} \). Similar to that of the Boolean network case, by Lemma 1 the equivalent algebraic form of probabilistic Boolean network (8) can be obtained as

\[
y(t + 1) = G(t)y(t),
\]

where \( y(t) = \phi_n (Y(t)) \in \Delta_2^n \), \( G(t) \in \{G_1, G_2, \ldots, G_s\} \) where \( G_i \in \mathcal{L}_{2^n \times 2^n} \) is the corresponding transition matrix of \( g_i \), and
\[ \text{Pr}(G(t) = G_i) = p_i, \quad i = 1, 2, \ldots, s. \] Before constructing a Lyapunov function for (9), we first give the definitions of stability and the Lyapunov function for (9) as follows.

**Definition 2:** Probabilistic Boolean network (9) is said to be stochastically stable at \( \delta_{2n}^n \) if \( \lim_{t \to \infty} E[y(t)] = \delta_{2n}^n \), where \( E[y(t)] \) represents the expectation of \( y(t) \).

**Definition 3:** A stochastic function \( V_2 : \Delta_{2n} \to \mathbb{R} \) is called a Lyapunov function for probabilistic Boolean network (9) if the following conditions hold:

- \( V_2(y(t)) > 0 \) for \( y(t) \neq \delta_{2n}^n \) and \( V_2(y(t)) = 0 \) for \( y(t) = \delta_{2n}^n \);
- \( \Delta V_2(y(t)) < 0 \) for \( y(t) \neq \delta_{2n}^n \) and \( \Delta V_2(y(t)) = 0 \) for \( y(t) = \delta_{2n}^n \), where \( \Delta V_2(y(t)) = E[V_2(y(t + 1))|y(t)] - V_2(y(t)) \).

**Lemma 2:** Based on Definitions 2 and 3, probabilistic Boolean network (9) is stochastically stable at \( \delta_{2n}^n \) if and only if there exists a Lyapunov function of network (9).

**Proof (Necessity):** By Definition 2 if probabilistic Boolean network (9) is stochastically stable at \( \delta_{2n}^n \), then for any initial state \( x(0) \), \( \lim_{t \to \infty} E[y(t)] = 1 \) and \( G_i \in \mathcal{L}_{2n \times 2n}, \quad i = 1, 2, \ldots, s \). Split \( G \) and \( E[y(t)] \) respectively, as

\[
G = \sum_{i=1}^s p_i G_i, \quad E[y(t)] = \left[ \begin{array}{c} \bar{y}_1(t) \\ \bar{y}_2(t) \end{array} \right],
\]

where \( G_{11} \in \mathbb{R}^{(2n-1)\times(2n-1)}, G_{22} \in \mathbb{R}, \bar{y}_2(t) \in \mathbb{R}^{2n-1} \) and \( \bar{y}_2(t) \in \mathbb{R} \). Thus, \( \lim_{t \to \infty} E[y(t)] = \delta_{2n}^n \) if and only if \( \lim_{t \to \infty} \bar{y}_1(t) = 0_{2n-1} \) and \( \lim_{t \to \infty} \bar{y}_2(t) = 1 \). Taking limitation on both side of (10) yields \( \lim_{t \to \infty} E[y(t + 1)] = G \lim_{t \to \infty} E[y(t)] \), which implies \( G_{22} = 1 \). Under this case where \( G_{22} = 1 \), the update of \( \bar{y}_1(t) \) can be written as

\[
\bar{y}_1(t + 1) = G_{11} \bar{y}_1(t),
\]

since \( G_{12} = 0_{2n-1} \) is implied by \( \sum_{i=1}^s p_i G_{12} G_{22} = 1 \) and \( G_{12} \geq 0 \). System (11) is a positive discrete system, then \( \lim_{t \to \infty} \bar{y}_1(t) = 0_{2n-1} \) if and only if there exists a vector \( \nu_1 \in \mathbb{R}^{2n-1} \) such that

\[
\nu_1^T \nu_1 > 0,
\]

\[
G_{11} \nu_1 - \nu_2 < 0.
\]

**Definition 4** ([33], [34]): Markovian switching Boolean network (16) is said to be stochastically stable at \( \delta_{2n}^n \) if for any initial value \( z(0) \) and any initial distribution of \( \sigma(t) \), the following condition holds:

\[
\lim_{t \to \infty} E[z(t)|z(0), \sigma(0)] = \delta_{2n}^n.
\]

For network (16), the Lyapunov function is defined as follows.

**Definition 5** ([33]): A stochastic function \( V_3 : \Delta_{2n} \times \mathbb{R} \to \mathbb{R} \) is called a Lyapunov function of network (16) if for any \( \sigma(t) \in \mathbb{R} \),

- \( V_3(z(t), \sigma(t)) > 0 \) for \( z(t) \neq \delta_{2n}^n \) and \( V_3(z(t), \sigma(t)) = 0 \) for \( z(t) = \delta_{2n}^n \);
- \( \Delta V_3(z(t), \sigma(t)) < 0 \) for \( z(t) \neq \delta_{2n}^n \) and \( \Delta V_3(z(t), \sigma(t)) = 0 \) for \( z(t) = \delta_{2n}^n \), where \( \Delta V_3(z(t), \sigma(t)) = E[V_3(z(t + 1), \sigma(t + 1))|z(t), \sigma(t)] - V_3(z(t), \sigma(t)) \).

Note that \( H_{\sigma(t)} = H_i \) when \( \sigma(t) = i \). Split \( H_i \) as

\[
H_i = \begin{bmatrix} H_{i,11} & H_{i,12} \\ H_{i,21} & H_{i,22} \end{bmatrix}, \quad H_{i,11} \in \mathbb{R}^{(2n-1)\times(2n-1)},
\]

for \( i = 1, 2, \ldots, r \). By recalling the stability results in ([33], [34]), one necessary condition for stochastic stability of network (16) is \( H_{i,22} = 1 \) for all \( i \in \mathbb{R} \). Then one Lyapunov function for Markovian switching Boolean network (16) can be designed as

\[
V_3(z(t), \sigma(t)) = \omega_1^T \sigma(t),
\]

where \( \omega_i \in \mathbb{R}^{2n}, \quad \omega_i = (\omega_{11}^T, 0)^T \in \mathbb{R}^{2n}, \quad i = 1, 2, \ldots, r \), satisfying

\[
\sum_{j=1}^r \pi_{ij} H_{i,11}^T \omega_{11} - \omega_{11} < 0,
\]

\[
\omega_{11} > 0.
\]

It has also been proved in ([33]) that Markovian switching Boolean network (16) is stochastically stable at \( \delta_{2n}^n \) if and only if there exists a Lyapunov function for (16) defined in Definition 5.

**IV. SELF-TRIGGERED SCHEDULING**

Due to the overuse of the resources and limited measurement conditions, we aim to design self-triggered strategy to control the system based on state measurements on the sampling times. In fact, the control strategy under self-triggered case has the structure as follows:

\[
\left\{ \begin{array}{l}
\ell(t) = u(t_k) \in U(x(t_k)), \quad t \in [t_k, t_{k+1}),
\ell_{k+1} = \ell_k + \tau(x(t_k)),
\end{array} \right.
\]

(22)
where \( t_0 = 0 \), \( \tau(x(t_k)) \) denotes the time between two successive sampling times, and \( U(x(t_k)) \) is the possible control set when the state is \( x(t_k) \). The problem we are interested in is to solve the co-design problem of both the triggering times and the required control.

In this section, self-triggered control for BCNs, probabilistic BCNs and Markovian switching BCNs is studied mainly based on the Lyapunov theory presented in the previous section. Hereafter, it is natural to assume that the studied BCNs can be (stochastically) stabilizable at the equilibrium point \( \delta^m_{2n} \) since the stability of BCNs can be viewed as a priori by the existing methods in [23, 40, 41].

A. BCNs

In this subsection, we just study the BCN from its algebraic expression form as

\[
x(t+1) = Fu(t)x(t),
\]

where \( x(t) \in \Delta^m_{2n} \) is the state variable, \( u(t) \in \Delta^m_{2n} \) is the control input and \( F \in \mathcal{L}^m_{2n\times2n+m} \). If BCN (23) is stabilizable at \( \delta^m_{2n} \) by a state feedback control \( u(t) = Kx(t) \),

\[
u(t) = Kx(t),
\]

where \( K \in \mathcal{L}^m_{2n\times2n} \), then by recalling the Lyapunov function in Subsection III-A, there exists a Lyapunov function \( V_1(x(t)) = \lambda^T x(t) \) for the closed-loop system

\[
x(t+1) = FK\Phi_{2n}x(t),
\]

satisfying

\[
\lambda = (\lambda_1, \ldots, \lambda_{2n})^T, \quad \lambda_1 > 0, \quad \tilde{F}_{11}^T\lambda_1 - \lambda_1 < 0,
\]

where \( \Phi_{2n} = \text{diag}(\delta_{2n}^m, \delta_{2n}^m, \ldots, \delta_{2n}^m) \) is called a reduced order matrix such that \( \Phi_{2n}x(t) = x(t) \otimes x(t) \), \( \tilde{F}_{11} = \begin{pmatrix} I_{2n-1} & 0_{2n-1} \\ 0_{2n-1} & 0_{x(2n-1)} \end{pmatrix} \)

By recalling the Lyapunov function in Subsection III-A, the self-triggered scheduling is designed such that the Lyapunov function at the next time will decrease. For \( M \geq 1 \), if for any \( t \) and any \( u \in \Delta^m_{2n} \), \( (Fu)^N x(t_k) \neq \delta^m_{2n} \), denote

\[
U_M(x(t_k)) = \{ u \in \Delta^m_{2n} | V_1((Fu)^i x(t_k)) - V_1((Fu)^{i-1} x(t_k)) < 0, i = 1, 2, \ldots, M \}.
\]

Then \( \tau(x(t_k)) \) and \( U(x(t_k)) \) are defined formally as

\[
\tau(x(t_k)) = \max \{ M \mid U_M(x(t_k)) \neq \emptyset \},
\]

\[
U(x(t_k)) = U_{\tau(x(t_k))}(x(t_k)).
\]

Proof. To show the well-definedness of the control strategy (22), it suffices to prove that for all \( x \in \Delta^m_{2n} \), \( U_0(x) \neq \emptyset \) where \( U_0(x) \) is defined in (26). Suppose that at some sampling time \( t_k \), \( x(t_k) = x \).

Choosing \( \bar{u} = Kx \), \( K \in \mathcal{L}^m_{2n\times2n} \), we have

\[
V_1(F\bar{u}x) - V_1(x) = V_1(FK\Phi_{2n}x(t_k)) - V_1(x(t_k)) = V_1(x(t_k + 1)) - V_1(x(t_k)) \begin{cases} = 0, & \text{if } \delta^m_{2n} \in \Phi_{2n}x(t_k) \\ < 0, & \text{if } x \neq \delta^m_{2n} \end{cases}
\]

by the definition of Lyapunov function \( V_1(x(t)) \). This proves that \( U_0(x) \neq \emptyset \), and thus \( t_{k+1} > t_k \).

Now we are in a position to prove that there exits a positive integer \( N < 2^n \) such that the update of the control \( u(t) \) stops at \( t_N \), i.e., \( t_N < \infty \) and \( t_{N+1} = \infty \). Bearing in mind the self-triggered scheduling in (25) and (29), under one designed self-triggered controller, we have \( V_1(x(t)) > V_1(x(t+1)) \) if \( x(t) \neq \delta^m_{2n} \) and \( V_1(x(t+1)) = V_1(x(t+1)) \). When \( N < 2^n \), then by the definition of the Lyapunov function \( V_1(x(t)) \), we can find an integer \( i \) satisfying \( 0 \leq i < N \) such that \( V_1(x(t)) = 0 \), i.e., \( x(t) = \delta^m_{2n} \). At time \( t_i \), by selecting \( u(t_i) = Kx(t_i) \), where \( K \in \mathcal{L}^m_{2n\times2n} \), then \( (Fu(t_i)) x(t_k) = x(t_k) \) for any \( t_i \geq 0 \). That is, for any \( t_i \), \( V_1(x(t)) = 0 \) for all \( t_i \). Then the control will not update after \( t_i \), i.e., \( t_{i+1} = \infty \), which is a contradiction to \( t_N < \infty \) and \( N > i \).

Next, we show that the system (23) with the control strategy (22) reaches the stable point \( \Phi_{2n}x(t) \) at a finite time and remains unchanged.

By the self-triggered condition, we have that \( V_1(t_0) > V_1(t_1) > \cdots > V_1(t_N) \geq V_1(t_{N+1}) = 0 \). If \( x(t_N) = \delta^m_{2n} \), then the constant control \( u(t_N) = Kx(t_N) \) can guarantee that the state of the system (23) is \( \delta^m_{2n} \) afterwards. That is to say the system (23) is stabilizable at \( \delta^m_{2n} \) in finite time \( t_N \). If \( x(t_N) \neq \delta^m_{2n} \), then the constant control \( u(t_N) = Kx(t_N) \) can guarantee that the state of the system (23) reaches \( \delta^m_{2n} \) in time \( 2^n \) since the control after \( t_N \) is invariant. Therefore the system (23) is stabilizable at \( \delta^m_{2n} \) in finite time \( t_N + 2^n \). The proof is completed.

From the above analysis, self-triggered controllers are not unique and can also be designed based on the decrease of the Lyapunov function. After \( t_{k+1} \) at which the control should be updated is determined, the state at time \( t_{k+1} \) and the possible control set \( U(x(t_{k+1})) \) can also be computed. In order to ensure the fast convergence at \( \delta^m_{2n} \), the control at time \( t_{k+1} \) can be chosen from the possible control set such that the Lyapunov function \( V(x(t)) \) takes the smallest value at \( t_{k+1} + 1 \). This control in our paper is called the optimal control. The detailed control design process is given as follows. Define

\[
I(x(t_{k+1})) = \arg \min_{u \in U(x(t_{k+1}))} \{ \lambda^T Fux(t_{k+1}) \}.
\]

Then the optimal self-triggered controller can be given as

\[
\begin{cases}
 u(t) = u(t_k), & \text{for } t \in [t_k, t_{k+1});
 u(t) = I(x(t_k+1)), & \text{for } t = t_{k+1}.
\end{cases}
\]

Example 1: Consider a BCN with \( n = 3, m = 1 \) and the transition matrix in (23) as

\[
F = \delta_{0}[2, 3, 3, 3, 7, 7, 8, 8, 4, 4, 6, 6, 8, 8, 5, 5].
\]

One feasible state feedback controller can be designed as \( K = [\delta_1, 1, 2, 2, 1, 2, 2, 1, 2] \). Then a Lyapunov function exists in the form \( V(x(t)) = \lambda^T x(t) \) with \( \lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \lambda_8)^T \) satisfying \( 0 < \lambda_5 < \lambda_7 < \lambda_6 < \lambda_3 < \lambda_2 < \lambda_1 \) and \( \lambda_6 < \lambda_4 \). Consequently, the self-triggered scheduling can be designed based on the Lyapunov stability theory. If at \( t_0 = 0 \), \( x(t_0) = \delta^m_2 \), one has \( V_1(x(t_0)) = \lambda_1 > V_1(Fu(t_0)x(t_0)) = \lambda_2 > \lambda_6 > \lambda_4 > \lambda_3 > \lambda_2 > \lambda_1 > \lambda_6 > \lambda_4 > \lambda_3 > \lambda_2 > \lambda_1 \).
\[ V_1((F(u(t))x(t_0))^2) = \lambda_3 = V_1((F(u(t_0))x(t_0))) \]
Then the next triggering time is \( t_1 = 2 \). By selecting \( u(t_1) = \delta_2^2 \), \( V_1(x(t_1)) = \lambda_3 > V_1((F(u(t_1))x(t_1))) = \lambda_0 \geq V_1((F(u(t_1)))x(t_1)) = \lambda_0 \). Then the next triggering time is \( t_2 = 4 \). Choosing \( u(t_2) = \delta_2^4 \) yields the next triggering time is \( t_3 = \infty \). Therefore, it only needs the sampled data and control strategies at three times \( t = 0, t = 2 \) and \( t = 4 \) to ensure the studied BCN stable at \( \delta_2^6 \). Note that the self-triggered sampling times are related with the initial states. Fortunately, we can design all the sampling times according the mechanism corresponding to different initial states as follows.

**TABLE I**

| Initial states | Sampling times | Control |
|----------------|----------------|---------|
| \( \delta_2^3 \) | \( t_0 = 0, t_1 = 2 \) | \( u(t_0) = \delta_2^3, u(t_1) = \delta_2^5 \) |
| \( \delta_2^4 \) | \( t_0 = 0, t_1 = 1 \) | \( u(t_0) = \delta_2^4, u(t_1) = \delta_2^2 \) |
| \( \delta_2^5 \) | \( t_0 = 0, t_1 = 2 \) | \( u(t_0) = \delta_2^5, u(t_1) = \delta_2^5 \) |
| \( \delta_2^6 \) | \( t_0 = 0, t_1 = 1 \) | \( u(t_0) = \delta_2^6, u(t_1) = \delta_2^2 \) |

### B. Probabilistic BCNs

Consider a probabilistic BCN as

\[
y(t + 1) = G(t)u(t)y(t),
\]

where \( y(t) \in \Delta_{2^n} \) is the state variable, \( u(t) \in \Delta_{2^n} \) is control input, and \( G(t) \in \{G_1, G_2, \ldots, G_s\} \) with \( G_i \in \mathcal{E}_{2^n \times 2^n} \). Moreover, \( \mathcal{P}(G(t) = G_i) = p_i \), where \( p_i > 0 \) and \( \sum_{i=1}^{s} p_i = 1 \). Assume that probabilistic BCN (31) is stabilizable by state feedback controller

\[
u = K(t)x(t),
\]

where \( K(t) \in \{K_1, K_2, \ldots, K_s\} \) with \( K_i \in \mathcal{E}_{2^n \times 2^n} \) and \( \mathcal{P}(K(t) = K_i) = p_i \), \( i = 1, 2, \ldots, s \). Then the closed-loop system

\[
y(t + 1) = G(t)(K(t)\Phi_2^n)y(t)
\]

is stochastically stable to \( \delta_{2^n}^2 \). By taking expectation on both sides of (32), one has

\[
\mathbb{E}\{y(t+1)\} = \sum_{i=1}^{s} p_i G_i K_i \Phi_2^n \mathbb{E}\{y(t)\} := \tilde{G} \mathbb{E}\{y(t)\},
\]

where \( \tilde{G} = \sum_{i=1}^{s} p_i G_i K_i \Phi_2^n \). Based on the Lyapunov function for probabilistic Boolean networks in Subsection II-B there exists a Lyapunov function \( V_2(y(t)) = \nu^T \nu(t) \) for the closed-loop system (33) satisfying

\[
\nu = (\nu_i^T, 0)^T, \quad \nu_i > 0, \quad \tilde{G}_{11}^{T} \nu_i - \nu_i < 0,
\]

where \( \tilde{G}_{11} = [I_{2^n-1} \quad 0_{2^n-1} \quad 0_{2^n-1}] \tilde{G} [I_{2^n-1} \quad 0_{2^n-1} \quad 0_{2^n-1}]^T \). Then the self-triggered scheduling (32) for probabilistic BCN (31) can be designed as follows. For \( M > 0 \), if for any \( u \in \Delta_{2^n} \) and any \( t \), \( \mathbb{E}\{y_{u,t}(t_k)\} \neq \delta_{2^n}^2 \), denote

\[
\mathcal{U}_M(y(t_k)) = \{u \in \Delta_{2^n} | \mathbb{E}\{V_2(y_{u,t_1}(t_k))\} \leq M \}
\]

where \( y_{u,t}(t_k) = (G(k+M-1)u \cdots (G_1u)y(t_k) \) and \( y_{u,0}(t_k) = y(t_k) \). Otherwise, if there exist some \( u \in \Delta_{2^n} \) and a positive integer \( N_u \leq M \) such that \( \mathbb{E}\{y_{u,N_u}(t_k)\} \neq \delta_{2^n}^2 \), denote

\[
\mathcal{U}_M(y(t_k)) = \{u \in \Delta_{2^n} | \mathbb{E}\{V_2(y_{u,t_1}(t_k))\} \leq M \}
\]

where \( y_{u,t}(t_k) = (G(k+M-1)u \cdots (G_1u)y(t_k) \) and \( y_{u,0}(t_k) = y(t_k) \). Otherwise, if there exist some \( u \in \Delta_{2^n} \) and a positive integer \( N_u \leq M \) such that \( \mathbb{E}\{y_{u,N_u}(t_k)\} \neq \delta_{2^n}^2 \), denote

\[
\mathcal{U}_M(y(t_k)) = \{u \in \Delta_{2^n} | \mathbb{E}\{V_2(y_{u,t_1}(t_k))\} \leq \infty \}
\]

Theorem 2: Consider probabilistic BCN (31). The control strategy in (32) for (31) is well defined, i.e., \( t_{k+1} > t_k \) for \( k = 1, 2, \ldots \). Moreover, the system (31) with the control strategy in (32) is stochastically stabilizable at \( \delta_{2^n}^2 \).

Proof. Similar to the proof of Theorem 1 it suffices to prove that for all \( y \in \Delta_{2^n}, \mathcal{U}_k(y) \neq \emptyset \). By taking expectation on both sides of (34),

\[
\mathbb{E}\{V_2(y_{u,1})\} - \mathbb{E}\{V_2(y_{u,t_1}(t_k))\}
\]

where the last inequality is implied by the Lyapunov stability theory for probabilistic Boolean networks. Thus, \( \tau(t_k) \geq 1 \) and the control strategy is well defined.

Now we will prove that the system (31) with the control strategy in (32) is stochastically stabilizable at \( \delta_{2^n}^2 \). In what follows, two cases are discussed.

Case 1: There is a minimal finite time \( N \) such that for any \( y(0) \in \Delta_{2^n}, \mathbb{E}\{y(N)|y(0)\} = \delta_{2^n}^2 \). Suppose that a maximal \( k \) can be found such that \( t_k \leq N \). Under the control \( u(t) = u_{t_k} \) for any \( t_k \leq t \leq N \), we have \( \mathbb{E}\{y(N)|y(t_k)\} = \delta_{2^n}^2 \). Based on the sampling scheduling, an appropriate control \( u \) can be selected such that for all \( t \geq N \), \( \mathbb{E}\{V_2(y_{u,t-N}(N))|y(N)\} = 0 \), which is equivalent to \( \mathbb{E}\{y_{u,t-N}(N)|y(N)\} = \delta_{2^n}^2 \). That is to say, the sampling time will stop at \( k \) or \( N \). At this point, the system (31) is stochastically stabilizable at \( \delta_{2^n}^2 \) in finite time.

Case 2: A finite time \( N \) satisfying that for any \( y(0) \in \Delta_{2^n}, \mathbb{E}\{y(N)|y(0)\} = \delta_{2^n}^2 \) cannot be found. Then for any time \( t \) and any \( y(0) \in \Delta_{2^n}, \mathbb{E}\{y(t)|y(0)\} = \delta_{2^n}^2 \) and \( V_2(y(t_0)) > 0 \). One result holds that for any \( k \), \( \mathbb{E}\{V_2(y(t_k+1)|y(t_k)) - V_2(y(t_k)) \leq 0 \), based on which a sufficiently small positive number \( \alpha < 1 \) can be found such that

\[
\mathbb{E}\{V_2(y(t_k+1)|y(t_k)) \leq (1-\alpha)V_2(y(t_k)) \}
\]

for any \( k = 0, 1, \ldots \). Taking expectation on both sides of (39) yields

\[
\mathbb{E}\{V_2(y(t_{k+1})|y(t_k)) \leq (1-\alpha)\mathbb{E}\{V_2(y(t_k))\} \}
\]

that is,

\[
\mathbb{E}\{V_2(y(t_{k+1})|y(t_k)) \leq (1-\alpha)^k\mathbb{E}\{V_2(y(t_0))\} \}
\]

By iteration,

\[
\mathbb{E}\{V_2(y(t_k)) \leq (1-\alpha)^k\mathbb{E}\{V_2(y(t_0))\} \}
\]

Making \( k \to \infty \) produces \( \lim_{k \to \infty} \mathbb{E}\{V_2(y(t_k))\} = 0 \), which is equivalent to \( \lim_{k \to \infty} \mathbb{E}\{y(t_k)\} = \delta_{2^n}^2 \).
Similar to BCN case, denote

$$I(y(t_{k+1})) = \arg \min_{u \in I(y(t_{k+1}))} \left\{ \nu_y^T \sum_{i=1}^n p_i G_i u(y(t_{k+1})) \right\}.$$  

Then the optimal self-triggered controller can be given as

$$\left\{ \begin{array}{l}
u(t) = u(t_k), \text{ for } t \in [t_k, t_{k+1}); \\
u(t) \in I(y(t_{k+1})), \text{ for } t = t_{k+1}. \end{array} \right.$$  \( (40) \)

Next we give an example on a probabilistic Boolean control network to show that its stochastic stability can be ensured by our self-triggered feedback control strategy.

**Example 2**: Consider a probabilistic BCN in the form of \( (31) \) with \( n = 3, m = 1 \), and \( \mathcal{P}\{G(t) = G_1\} = p_1 = 0.3, \mathcal{P}\{G(t) = G_2\} = p_2 = 0.7 \), where \( G_1 = \delta_{8}[3, 1, 6, 2, 2, 2, 8, 8, 1, 1, 1, 8, 4, 3, 5, 8] \), \( G_2 = \delta_{8}[1, 1, 2, 6, 8, 7, 7, 6, 1, 1, 5, 5, 8] \).

Using the existing state feedback control design method in \( [39] \), one feasible update-based feedback control can be given as \( \mathcal{P}\{K(t) = K_1\} = 0.3 \) and \( \mathcal{P}\{K(t) = K_2\} = 0.7 \), where

$$K_1 = \delta_{2}[1, 1, 1, 2, 2, 2, 1],$$
$$K_2 = \delta_{2}[2, 2, 1, 1, 1, 2, 2].$$

Then one feasible Lyapunov function can be given as \( V_2(y(t)) = \nu_y^T y(t) \), where \( \nu = (8.3, 9.3, 9.4, 5.5, 2.6, 6.4, 3.6, 0)^T \). Via the obtained results in this subsection, the simulation results are shown in Figure 1. In Figure 1(a), we take the initial state \( y(0) = \delta_{8}^{11} \) and the corresponding state trajectories are shown by running the program 500 times. In Figure 1(b), the possible trajectories corresponding to all initial states are simulated. From these, it can also be seen that the stochastic stability at \( \delta_{8}^{11} \) can be ensured.

On the other hand, the stochastic stability defined in Definition \( [39] \) is different from the stability defined in \( [41] \), which is always called finite-time stability (in \( 2^n \)). However, the stochastic stability defined in this paper may not be achieved in finite time.

### C. Markovian switching BCNs

Consider a Markovian switching BCN as

$$z(t + 1) = H_{\sigma(t)} u(t) z(t),$$  \( (41) \)

where \( z(t) \in \Delta^{2^n} \) is the state variable, \( u(t) \in \Lambda^{2_m} \) is the control input, \( \sigma(t) \) is the switching signal, and \( H_{\sigma(t)} \in \{ H_1, H_2, \ldots, H_r \} \) with \( H_i \in \mathcal{L}_{2^n \times 2^n}, i = 1, 2, \ldots, r \). Here \( \sigma(t) \) is a discrete Markov chain same as in Subsection III-C. If Markovian switching BCN \( (41) \) is stochastically stabilizable at \( \delta_{2^n}^{11} \) by a state feedback control

$$u(t) = K_{\sigma(t)} x(t),$$  \( (42) \)

where \( K_{\sigma(t)} \in \{ K_1, K_2, \ldots, K_r \} \) with \( K_i \in \mathcal{L}_{2^n \times 2^n}, i = 1, 2, \ldots, r \), then the closed-loop system

$$z(t + 1) = H_{\sigma(t)} K_{\sigma(t)} \Phi_{2^n} z(t)$$  \( (43) \)

is stochastically stable at \( \delta_{2^n}^{11} \). Based on the Lyapunov function for Markovian switching Boolean networks in Subsection III-C there exists a Lyapunov function \( V_2(z)(t), \sigma(t) = \omega_{\sigma(t)}^T z(t) \) for the closed-loop system \( (43) \) satisfying for \( i = 1, 2, \ldots, r \), \( \omega_{11} > 0 \)

$$\sum_{j=1}^{p} \pi_{ij} H_{i,11}^T \omega_{j1} - \omega_{11} < 0,$$  \( (44) \)

Then \( \tau(z(t_k), \sigma(t_k)) \) and \( U(z(t_k), \sigma(t_k)) \) are defined as

$$\tau(z(t_k), \sigma(t_k)) = \max\{ M \mid U_M(z(t_k), \sigma(t_k)) \neq \emptyset \},$$  \( (45) \)
$$U(z(t_k), \sigma(t_k)) = U_{c(z(t_k), \sigma(t_k))}(z(t_k), \sigma(t_k)).$$  \( (46) \)
Theorem 3: Consider Markovian switching BCN (11). The control strategy in (11) is well defined, i.e., \( t_{k+1} > t_k \) for \( k = 1, 2, \ldots \). Also the system (11) is stochastically stabilizable at \( \delta_2^n \).

Proof. Similar to the proof of Theorem 1, we only need to prove for all \( z \in \Delta^n \) and \( s \in \mathcal{S} \), there exists one \( \bar{u} \in \Delta^n \) such that \( \bar{u}(z, i) \). Suppose that at some time \( t_k \), \( z(t_k) = z \). Let \( \bar{u} = K_\sigma(t_k)z \), where \( K_\sigma(t_k) \) is the stabilizing controller given in (9) and by the properties of Lyapunov function in Subsection III-C and similar to the proof of Theorem 2, it is easy to get that \( E\{V_3(z(t_k + 1))|\sigma(t_k), \sigma(t_1)\} - V_3(z(t_k), \sigma(t_k)) = 0 \) if \( z \in \bar{\delta}_2^n \) and \( E\{V_3(z(t_k + 1))|\sigma(t_k), \sigma(t_1)\} - V_3(z(t_k), \sigma(t_k)) < 0 \) if \( z \notin \bar{\delta}_2^n \), which implies that \( t_{k+1} > t_k \).

Similar to the proof of Theorem 2 the final statement can also be proved.

Denote \( \mathcal{I}(z(t_{k+1}), \sigma(t_{k+1})) = \{ \}
\]
\[
\eta \in \mathcal{E}(z(t_{k+1}), \sigma(t_{k+1})) = \left\{ \sum_{j=1}^r \pi_{ij} \omega_j^T H_i \omega_j(z(t_k)) \right\}. 
\]
Then the optimal self-triggered controller can be given as
\[
\begin{align*}
& u(t) = u(t_k), \quad \text{for } t \in [t_k, t_{k+1}); \\
& u(t) = \mathcal{I}(z(t_{k+1}), \sigma(t_{k+1})), \quad \text{for } t = t_{k+1}.
\end{align*}
\]

D. Discussions

We have designed a type of self-triggered scheduling for different kinds of BCNs, including deterministic BCNs, probabilistic BCN and Markovian switching BCNs, respectively, by reviewing and proposing the Lyapunov stability theory for the corresponding Boolean networks. For deterministic Boolean networks, a deterministic function, of course, can be regarded as a Lyapunov function. The Lyapunov functions for probabilistic and Markovian switching Boolean networks are both stochastic, while the one for a probabilistic Boolean network can be equipped with a common gain \( \nu \) and the one for a Markovian switching Boolean network is in fact composed of multiple functions. It is also difficult to find a common Lyapunov function for a Markovian switching Boolean network.

It can been seen that state feedback controllers are a special case of the self-triggered controllers by regarding all the time duration between the successive sampling times as 1.

As analyzed in the previous subsections, the definition of stochastic stability for stochastic Boolean networks, i.e., probabilistic and stochastic Boolean networks, can be given as follows (taking Markovian switching Boolean network (16) as an example).

Definition 6: Markovian switching Boolean network (16) is said to be stochastically stable at \( \delta_2^n \) in finite time if for any initial value \( z(0) \) and any initial distribution of \( \sigma(t) \), there exists an integer \( T > 0 \) such that for all \( t > T \), one has
\[
E\{z(t)|z(0), \sigma(0)\} = \delta_2^n.
\]

With the stochastic stability defined as above, which is also called finite-time stochastic stability, an integer \( 0 < N < 2^n \) can be found such that the control will not update after time \( t_N \).

V. CONCLUSION

In this paper, we studied self-triggered control for three kinds of BCNs, including deterministic, probabilistic and Markovian switching BCNs, in order to reduce the computational loads and deal with the constraint of limited resources. By first reviewing or proposing Lyapunov stability theory for Boolean networks, the self-triggered scheduling was designed based on the decrease of the Lyapunov function between two successive samplings and the self-triggered controller was designed, under which the studied BCNs can be ensured to be stabilizable at \( \delta_2^n \).

REFERENCES

[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] M. H. Kabir, M. R. Hoque, B. J. Koo, and S. H. Yang, “Mathematical modelling of a context-aware system based on Boolean control networks for smart home,” in The 8th IEEE International Symposium on Consumer Electronics (ISCE 2014), 2014, pp. 1–2.
[3] J. M. Alexander, “Random boolean networks and evolutionary game theory,” Philosophy of Science, vol. 70, no. 5, pp. 1289–1304, 2003.
[4] D. Cheng, “On finite potential games,” Automatica, vol. 50, no. 7, pp. 1793–1801, 2014.
[5] X. Zhang, Y. Hao, and D. Cheng, “Incomplete-profile potential games,” Journal of The Franklin Institute, vol. 355, no. 2, pp. 862–877, 2018.
[6] D. Cheng and H. Qi, Semi-Tensor Product of Matrices — Theory and Applications. Beijing: Science Press, 2007.
[7] D. Cheng, H. Qi, and Z. Li, Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach. Springer, 2011.
[8] D. Cheng, Y. Zhao, and T. Xu, “Receding horizon based feedback optimization for max-valued logical networks,” IEEE Transactions on Automatic Control, vol. 60, no. 12, pp. 3362–3366, 2015.
[9] E. Fornasini and M. E. Valcher, “Observability, reconstructibility and state observers of Boolean control networks,” IEEE Transactions on Automatic Control, vol. 58, no. 6, pp. 1390–1401, 2013.
[10] Y. Guo, Y. Ding, and D. Xie, “Invariant subset and set stability of Boolean networks under arbitrary switching signals,” IEEE Transactions on Automatic Control, vol. 62, no. 8, pp. 4209–4214, 2017.
[11] H. Li, L. Xie, and Y. Wang, “On robust control invariance of Boolean control networks,” Automatica, vol. 68, pp. 392–396, 2016.
[12] J. Liang, H. Chen, and J. Lam, “An improved criterion for controllability of Boolean control networks,” IEEE Transactions on Automatic Control, vol. 62, no. 11, pp. 6012–6018, 2017.
[13] Y. Liu, B. Li, H. Chen, and J. Cao, “Function perturbations on singular Boolean networks,” Automatica, vol. 84, pp. 36–47, 2017.
[14] J. Lu, J. Zhong, C. Huang, and J. Cao, “On pinning controllability of Boolean control networks,” IEEE Transactions on Automatic Control, vol. 61, no. 6, pp. 1658–1663, 2016.
[15] M. Meng, J. Lam, J. Feng, and K. C. Cheung, “Stability and guaranteed cost analysis of time-triggered Boolean networks,” IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 8, pp. 3893–3899, 2018.
[16] M. Meng, G. Xiao, C. Zhai, and G. Li, “Controllability of Markovian jump Boolean control networks,” Automatica, vol. 106, pp. 70–76, 2019.
[17] 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, p. 010202, 2018.
[18] Y. Liu, J. Cao, L. Sun, and J. Lu, “Sampled-data state feedback stabilization of Boolean control networks,” Neural Computation, vol. 28, no. 4, pp. 778–799, 2016.
[19] S. Zhu, Y. Liu, J. Lou, J. Lu, and F. E. Alsaadi, “Sampled-data state feedback control for the set stabilization of Boolean control networks,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, doi: 10.1109/TSMC.2018.2852703.
[20] Y. Liu, L. Tong, J. Lou, J. Lu, and J. Cao, “Sampled-data control for the synchronization of boolean control networks,” IEEE Transactions on Cybernetics, vol. 49, no. 2, pp. 726–732, 2019.
[21] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks,” IEEE Transactions on Automatic Control, vol. 52, no. 9, pp. 1680–1685, 2007.
[22] W. Hu, L. Liu, and G. Feng, “Consensus of linear multi-agent systems by distributed event-triggered strategy,” IEEE Transactions on Cybernetics, vol. 46, no. 1, pp. 148–157, 2016.
[23] W. Xia, D. W. C. Ho, L. Li, and J. Cao, “Event-triggered schemes on leader-following consensus of general linear multiagent systems under different topologies,” IEEE Transactions on Cybernetics, vol. 47, no. 1, pp. 212–223, 2017.
[24] B. Li, Y. Liu, K. I. Kou, and L. Yu, “Event-triggered control for the disturbance decoupling problem of Boolean control networks,” IEEE Transactions on Cybernetics, vol. 48, no. 9, pp. 2764–2769, 2018.
[25] Y. Li, H. Li, and W. Sun, “Event-triggered control for robust set stabilization of logical control networks,” Automatica, vol. 95, pp. 556–560, 2018.
[26] Q. Zhu and W. Lin, “Stabilizing Boolean networks by optimal event-triggered feedback control,” *Systems & Control Letters*, vol. 126, pp. 40–47, 2019.

[27] J. Yang, J. Lu, L. Li, Y. Liu, Z. Wang, and F. E. Alsaadi, “Event-triggered control for the synchronization of Boolean control networks,” *Nonlinear Dynamics*, vol. 96, no. 2, pp. 1335–1344, 2019.

[28] L. Tong, Y. Liu, Y. Li, J. Lu, Z. Wang, and F. E. Alsaadi, “Robust control invariance of probabilistic Boolean control networks via event-triggered control,” *IEEE Access*, vol. 6, pp. 37767–37774, 2018.

[29] X. Wang and M. D. Lemmon, “Self-triggered feedback control systems with finite-gain $L_2$ stability,” *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 452–467, 2009.

[30] M. Mazo Jr, A. Anta, and P. Tabuada, “An ISS self-triggered implementation of linear controllers,” *Automatica*, vol. 46, no. 8, pp. 1310–1314, 2010.

[31] R. Li, M. Yang, and T. Chu, “State feedback stabilization for probabilistic Boolean networks,” *Automatica*, vol. 50, no. 4, pp. 1272–1278, 2014.

[32] F. Li and L. Xie, “Set stabilization of probabilistic Boolean networks using pinning control,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 8, pp. 2555–2561, 2019.

[33] M. Meng, L. Liu, and G. Feng, “Stability and $l_1$ gain analysis of Boolean networks with Markovian jump parameters,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4222–4228, 2017.

[34] M. Meng, J. Lam, J. Feng, and K. C. Cheung, “Stability and stabilization of Boolean networks with stochastic delays,” *IEEE Transactions on Automatic Control*, vol. 64, no. 2, pp. 790–792, 2019.

[35] Y. Wang and H. Li, “On definition and construction of Lyapunov functions for Boolean networks,” in *Proceedings of the 10th World Congress on Intelligent Control and Automation (WCICA)*, 2012, pp. 1247–1252.

[36] S. Liu and G. Trenkler, “Hadamard, Khatri-Rao, Kronecker and other matrix products,” *International Journal of Information and Systems Sciences*, vol. 4, no. 1, pp. 160–177, 2008.

[37] H. Li and Y. Wang, “Lyapunov-based stability and construction of Lyapunov functions for Boolean networks,” *SIAM Journal on Control and Optimization*, vol. 55, no. 6, pp. 3437–3457, 2017.

[38] M. Meng, J. Lam, J. Feng, and X. Li, “$l_1$-gain analysis and model reduction problem for Boolean control networks,” *Information Sciences*, vol. 348, pp. 68–83, 2016.

[39] S. Zhu, Q. Han, and C. Zhang, “$l_1$-gain performance analysis and positive filter design for positive discrete-time Markov jump linear systems: A linear programming approach,” *Automatica*, vol. 50, no. 8, pp. 2098–2107, 2014.

[40] R. Li, M. Yang, and T. Chu, “State feedback stabilization for Boolean control networks,” *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1853–1857, 2013.

[41] L. Wang, Y. Liu, Z. Wu, J. Lu, and L. Yu, “Stabilization and finite-time stabilization of probabilistic Boolean control networks,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, doi: 10.1109/TSMC.2019.2898880.