Optimal Control of Boolean Control Networks

Article *in* IEEE Transactions on Automatic Control · May 2014

Impact Factor: 2.78 · DOI: 10.1109/TAC.2013.2294821

Maria Elena Valcher
University of Padova

Optimal control of Boolean control networks

Ettore Fornasini and Maria Elena Valcher

Abstract— **In this paper we address the optimal control problem for Boolean control networks (BCNs). We first consider the problem of finding the input sequences that minimize a given cost function over a finite time horizon. The problem solution is obtained by means of a recursive algorithm that represents the analogue for BCNs of the difference Riccati equation for linear systems. We prove that a significant number of optimal control problems for BCNs can be easily reframed into the present set-up. In particular, the cost function can be adjusted so as to include penalties on the switchings, provided that we augment the size of the BCN state variable. In the second part of the paper, we address the infinite horizon optimal control problem and we provide necessary and sufficient conditions for the problem solvability. The solution is obtained as the limit of the solution over the finite horizon $[0, T]$, and it is always achieved in a finite number of steps. Finally, the average cost problem over the infinite horizon, investigated in [30], is addressed by making use of the results obtained in the previous sections.**

I. Introduction

Boolean networks (BNs) have recently witnessed a renewed interest, as they constitute an effective tool for representing a number of phenomena whose describing variables display only two operation levels. This is the case of genetic regulation networks [17], that can be successfully modeled as BNs, due to the fact that each gene can be naturally described as a binary device that displays only two states: "transcribed" or "quiescent", namely "on" or "off". BNs have also been used to describe the interactions among agents, and hence to investigate consensus problems [15].

Modeling a biological system involves considerable uncertainty, due to perturbations that affect the biological system and inaccuracies of the measuring equipment. Incorporating this uncertainty in the modeling stage leads to Probabilistic Boolean Networks (PBNs) [28]. A PBN is a collection of (deterministic) BNs combined with a probabilistic switching rule determining which network is active at each time instant. In this set-up, the optimal control problem, applied to genetic networks, has been the object of an extensive literature. Indeed, gene regulatory systems have been modeled via PBNs, and both optimal finite horizon and infinite horizon control policies have been proposed (see, e.g. [9], [25], [26], [27], [29]).

On the other hand, many biological systems have exogenous inputs and it is natural to extend BNs to Boolean Control Networks (BCNs) by adding Boolean inputs. In this respect, a BCN can be seen as a family of BNs, each of them associated with a specific value of the input variables. For example, when modeling the progression of a disease, a binary input may represent whether a certain medicine is administered or not at each time step, and BCNs with inputs have been used to design and analyze therapeutic intervention strategies. The idea is to find a control sequence that steers the network from an undesirable location (corresponding to a diseased state of the biological system) to a desirable one (corresponding to a healthy state).

In the last decade, an algebraic framework has been developed that casts both BNs and BCNs into the framework of linear state-space models (operating on canonical vectors) [3], [6], [7], [8]. Within this setting, several control problems, like stability, stabilizability [4], [13], [14], controllability [20] and observability [12], have been investigated.

The optimal control of BCNs has been recently addressed in a few contributions. Specifically, in [30] (see also Chapter 15 in [8]) the problem of finding the input sequence that maximizes, on the infinite horizon, an average payoff that weights both the state and the input at every time $t \in \mathbb{Z}_+$, was investigated. The optimal solution is expressed as a feedback law, driving the BCN to a periodic state trajectory with maximum payoff. Also, [18] and [19] considered the optimal control problem over a finite horizon, but restricted the analysis to the case when the payoff function only depends on the state of the BCN at the end of the control interval. The optimal solution is obtained by resorting to the maximum principle, and has the structure of a time varying state feedback law.

The interest in the optimal control problem for BCNs arises primarily, but not exclusively, from two research areas. On the one hand, interesting applications to game theory have been illustrated in [8], [30]. On the other hand, the results presented in [9], [25], [26], [27], [29] provide evidence of the fact that optimal control problems naturally arise when dealing with biological systems in general, and genetic networks in particular, and BCNs often represent a very convenient set-up where to investigate these problems. As an example, we consider the constrained intervention in a mammalian cell-cycle network [10], [11], [16].

Motivating Example: The cell cycle is a temporal sequence of molecular events that take place in a cell, leading to its division and duplication. The cell cycle is divided into several phases. DNA replication occurs during the Synthesis (or S) phase. Growth stops and cellular energy is focused on the orderly division into two daughter cells at the Mitosis (or M) phase. The S and M phases are separated by two gap phases, G1 (between M and S) and G2 (between S and M). A fifth phase, called G0, corresponding to a quiescent state, can be reached from G1 in the absence of stimulation. Gap phases enable the cell to monitor its environment and internal state before committing to the S or M phase.

Mammalian cell division is tightly controlled, and cell cycle

Ettore Fornasini and Maria Elena Valcher are with the Dipartimento di Ingegneria dell'Informazione, Università di Padova, via Gradenigo 6/B, 35131 Padova, Italy, {fornasini,meme}@dei.unipd.it.

A preliminary version of the first part of this paper has been accepted for presentation to IEEE CDC 2013.

coordination is achieved through extra-cellular positive and negative signals whose balance decides whether a cell will divide or remain in the G0 resting phase. The positive signals or growth factors ultimately elicit the activation of Cyclin D (CycD) in the cell.

[11] developed a BCN model for the core network regulating the mammalian cell cycle. The model includes a single Boolean input, $U(t)$, corresponding to the activation/inactivation of CycD in the cell, and nine Boolean state-variables $X_1(t), \dots, X_9(t)$ representing the activity/inactivity at time t of nine different proteins: Rb, E2F, CycE, CycA, p27, Cdc20, Cdh1, UbcH10, CycB, respectively. The BCN model is

$$\begin{aligned}
X_1(t+1) &= (\bar U(t) \wedge \bar X_3(t) \wedge \bar X_4(t) \wedge \bar X_9(t)) \\
&\quad \vee (X_5(t) \wedge \bar U(t) \wedge \bar X_9(t)), \\
X_2(t+1) &= (\bar X_1(t) \wedge \bar X_4(t) \wedge \bar X_9(t)) \\
&\quad \vee (X_5(t) \wedge \bar X_1(t) \wedge \bar X_9(t)), \\
X_3(t+1) &= X_2(t) \wedge \bar X_1(t), \\
X_4(t+1) &= (X_2(t) \wedge \bar X_1(t) \wedge \bar X_6(t) \wedge (\overline{X_7(t) \wedge X_8(t)})) \\
&\quad \vee (X_4(t) \wedge \bar X_1(t) \wedge \bar X_6(t) \wedge (\overline{X_7(t) \wedge X_8(t)})), \\
X_5(t+1) &= (\bar U(t) \wedge \bar X_3(t) \wedge \bar X_4(t) \wedge \bar X_9(t)) \\
&\quad \vee (X_5(t) \wedge (\overline{X_3(t) \wedge X_4(t)}) \wedge \bar U(t) \wedge \bar X_9(t)), \\
X_6(t+1) &= X_9(t), \\
X_7(t+1) &= (\bar X_4(t) \wedge \bar X_9(t)) \vee X_6(t) \vee (X_5(t) \wedge \bar X_9(t)), \\
X_8(t+1) &= \bar X_7(t) \\
&\quad \vee (X_7(t) \wedge X_8(t) \wedge (X_6(t) \vee X_4(t) \vee X_9(t))), \\
X_9(t+1) &= \bar X_6(t) \wedge \bar X_7(t),
\end{aligned} \tag{1}$$

where $\vee, \wedge$ and $\bar\cdot$ represent the logical functions OR, AND and NOT.

Following one of the mutations proposed in [11], in [10] it was assumed that, in a cancerous scenario, the gene p27 can mutate so that it is always inactive (OFF). This mutation leads to a situation where both CycD and Rb are down-regulated as "undesirable states". In this contest, the control strategy proposed in [10] aimed at preventing that the system evolves passing through configurations with both CycD and Rb set to zero. To this end the Authors introduced a function rewarding all the allotted configurations (and hence penalizing all the forbidden ones) and discussed a control strategy aimed at maximizing the reward function. We refer the reader to [10] for the details.

By following this stream of research, in this paper we address the optimal control problem for BCNs, by assuming a cost function that depends on both the state and the input values at every time instant. We first consider the finite horizon optimal control problem. By resorting to the semi-tensor product, the original cost function is rewritten as a linear one, and the problem solution is obtained, in section III, by means of a recursive algorithm that represents the analogue for BCNs of the difference Riccati equation for linear systems. In section IV, several optimal control problems for BCNs are shown to be easily reframed into the present set-up. In particular, in section V, the cost function is adjusted so as to include penalties on the switchings, provided that the size of the BCN state variable is suitably augmented.

In section VI, we address the infinite horizon optimal control problem and we provide necessary and sufficient conditions for its solvability. The solution is obtained as the limit of the solution over the finite horizon $[0, T]$, and it is always achievable in a finite number of steps. Finally, the average cost problem over the infinite horizon, investigated in [30], [8], is addressed, by making use of the results obtained in the previous sections.

Notation. $\mathbb{Z}_+$ denotes the set of nonnegative integers. Given two integers $k, n \in \mathbb{Z}_+$, with $k \le n$, by the symbol $[k, n]$ we denote the set of integers $\{k, k+1, \dots, n\}$. We consider Boolean vectors and matrices, taking values in $\mathcal{B} := \{0, 1\}$, with the usual Boolean operations.

δ^i_k denotes the ith canonical vector of size k, $\mathcal{L}_k$ the set of all k-dimensional canonical vectors, and $\mathcal{L}_{k\times n} \subset \mathcal{B}^{k\times n}$ the set of all $k \times n$ matrices whose columns are canonical vectors of size k. Any matrix $L \in \mathcal{L}_{k\times n}$ can be represented as a row whose entries are canonical vectors in $\mathcal{L}_k$, namely $L = [\, \delta^{i_1}_k \quad \delta^{i_2}_k \quad \dots \quad \delta^{i_n}_k \,]$, for suitable indices $i_1, i_2, \dots, i_n \in [1, k]$. The k-dimensional vector with all entries equal to 1, is denoted by $\mathbf{1}_k$.

The (ℓ, j)th entry of a matrix L is denoted by $[L]_{\ell,j}$, while the ℓth entry of a vector $\mathbf{v}$ is $[\mathbf{v}]_\ell$. The ith column of a matrix L is $\mathrm{col}_i(L)$. Given a matrix $L \in \mathcal{B}^{k\times k}$ (in particular, $L \in \mathcal{L}_{k\times k}$), we associate with it [2] a *digraph* $\mathcal{D}(L)$, with vertices $1, \dots, k$. There is an arc (j, ℓ) from j to ℓ if and only if the (ℓ, j)th entry of L is unitary. A sequence $j_1 \to j_2 \to \dots \to j_r \to j_{r+1}$ in $\mathcal{D}(L)$ is a *path* of length r from j_1 to j_{r+1} provided that $(j_1, j_2), \dots, (j_r, j_{r+1})$ are arcs of $\mathcal{D}(L)$. A closed path is called a *cycle*. In particular, a cycle γ with no repeated vertices is called *elementary,* and its length $|\gamma|$ coincides with the number of (distinct) vertices appearing in it.

There is a bijective correspondence between Boolean variables $X \in \mathcal{B}$ and vectors $\mathbf{x} \in \mathcal{L}_2$, defined by the relationship

$$\mathbf{x} = \begin{bmatrix} X \\ \bar X \end{bmatrix}.$$

We introduce the *(left) semi-tensor product* $\ltimes$ between matrices (in particular, vectors) as follows [8], [20], [22]: given $L_1 \in \mathbb{R}^{r_1\times c_1}$ and $L_2 \in \mathbb{R}^{r_2\times c_2}$ (in particular, $L_1 \in \mathcal{L}_{r_1\times c_1}$ and $L_2 \in \mathcal{L}_{r_2\times c_2}$), we set

$$L_1 \ltimes L_2 := (L_1 \otimes I_{T/c_1})(L_2 \otimes I_{T/r_2}), \qquad T := \text{l.c.m.}\{c_1, r_2\},$$

where l.c.m. denotes the least common multiple. The semi-tensor product represents an extension of the standard matrix product, by this meaning that if $c_1 = r_2$, then $L_1 \ltimes L_2 = L_1 L_2$. Note that if $\mathbf{x}_1 \in \mathcal{L}_{r_1}$ and $\mathbf{x}_2 \in \mathcal{L}_{r_2}$, then $\mathbf{x}_1 \ltimes \mathbf{x}_2 \in \mathcal{L}_{r_1 r_2}$. For the various properties of the semi-tensor product we refer to [8]. By resorting to the semi-tensor product, we can extend the previous correspondence to a bijective correspondence between $\mathcal{B}^n$ and $\mathcal{L}_{2^n}$. This is possible in the following way: given $X = [\, X_1 \quad X_2 \quad \dots \quad X_n \,]^\top \in \mathcal{B}^n$, set

$$\mathbf{x} := \begin{bmatrix} X_1 \\ \bar X_1 \end{bmatrix} \ltimes \begin{bmatrix} X_2 \\ \bar X_2 \end{bmatrix} \ltimes \dots \ltimes \begin{bmatrix} X_n \\ \bar X_n \end{bmatrix} = \begin{bmatrix} X_1 X_2 \dots X_{n-1} X_n \\ X_1 X_2 \dots X_{n-1} \bar X_n \\ X_1 X_2 \dots \bar X_{n-1} X_n \\ \vdots \\ \bar X_1 \bar X_2 \dots \bar X_{n-1} \bar X_n \end{bmatrix}.$$

II. Finite horizon optimal control of BCNs: problem statement

A *Boolean control network* is described by the following equation

$$X(t+1) = f(X(t), U(t)), \qquad t \in \mathbb{Z}_+, \tag{2}$$

where $X(t)$ and $U(t)$ denote the n-dimensional state variable and the m-dimensional input at time t, taking values in $\mathcal{B}^n$ and $\mathcal{B}^m$, respectively. f is a (logic) function, i.e. $f : \mathcal{B}^n \times \mathcal{B}^m \to \mathcal{B}^n$. By resorting to the semi-tensor product $\ltimes$, state and input Boolean variables can be represented as canonical vectors in $\mathcal{L}_N$, $N := 2^n$, and $\mathcal{L}_M$, $M := 2^m$, respectively, and the BCN (2) satisfies [8] the following *algebraic description*:

$$\mathbf{x}(t+1) = L \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad t \in \mathbb{Z}_+, \tag{3}$$

where $\mathbf{x}(t) \in \mathcal{L}_N$ and $\mathbf{u}(t) \in \mathcal{L}_M$. $L \in \mathcal{L}_{N \times NM}$ is a matrix whose columns are canonical vectors of size N. For every choice of the input variable at time t, namely for every $\mathbf{u}(t) = \delta_M^i$, $L \ltimes \mathbf{u}(t) =: L_i$ is a matrix in $\mathcal{L}_{N\times N}$. So, we can think of the BCN (3) as a switched Boolean system (see [21], [24] for some recent works on this subject; also switched BCNs have been recently investigated in [23]),

$$\mathbf{x}(t+1) = L_{\mathbf{u}(t)}\mathbf{x}(t), \qquad t \in \mathbb{Z}_+, \tag{4}$$

where $\mathbf{u}(t), t \in \mathbb{Z}_+$, is a switching sequence taking values in $[1, M]$. For every $i \in [1, M]$, we refer to the BN

$$\mathbf{x}(t+1) = L_i \mathbf{x}(t), \qquad t \in \mathbb{Z}_+, \tag{5}$$

as to the ith subsystem of the switched Boolean system. Note that the matrix L can be expressed in terms of the matrices L_i as:

$$L = [\, L_1 \quad L_2 \quad \dots \quad L_M \,].$$

By referring to the algebraic description of the BCN (3), we introduce the following general finite horizon optimal control problem:

Given the BCN (3), with initial state $\mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{L}_N$, determine an input sequence that minimizes the cost function:

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathcal{Q}_f(\mathbf{x}(T)) + \sum_{t=0}^{T-1} \mathcal{Q}(\mathbf{u}(t), \mathbf{x}(t)), \tag{6}$$

where $\mathcal{Q}_f(\cdot)$ is any function defined on $\mathcal{L}_N$, and $\mathcal{Q}(\cdot,\cdot)$ is any function defined on $\mathcal{L}_M \times \mathcal{L}_N$.

The cost function (6) weights the BCN state at every time instant: the final state is weighted by a special function $\mathcal{Q}_f(\cdot)$, while the state at every intermediate instant t is weighted, together with the input value at the same time, by the function $\mathcal{Q}(\cdot,\cdot)$. Note that $\mathcal{Q}(\mathbf{u}(t),\cdot)$ can also be thought of as $\mathcal{Q}_{\mathbf{u}(t)}(\cdot)$, and hence the cost function can be regarded as a "switched function", that depends on the specific "switched sequence" $\mathbf{u}(t), t \in [0, T-1]$ (see [31], Problem 1, where the quadratic optimal control problem of a discrete time switched linear system is investigated).

The first step to take is to show that, due to the fact that the state and input vectors are always canonical vectors, every cost function described as in (6) can be equivalently expressed as a linear cost function, by resorting to the semi-tensor product. Indeed, the final cost function can be equivalently expressed as

$$\mathcal{Q}_f(\mathbf{x}(T)) = \mathbf{c}_f^\top \mathbf{x}(T).$$

for $\mathbf{c}_f^\top := [\, \mathcal{Q}_f(\delta_N^1) \quad \mathcal{Q}_f(\delta_N^2) \quad \dots \quad \mathcal{Q}_f(\delta_N^N) \,]$. Similarly, by making use of the semi-tensor product properties, and of the fact that $\mathbf{x}(t) \in \mathcal{L}_N$ and $\mathbf{u}(t) \in \mathcal{L}_M$, for every $t \in [0, T-1]$, we have that

$$\mathcal{Q}(\mathbf{u}(t), \mathbf{x}(t)) = \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t),$$

where

$$\begin{aligned}\mathbf{c}^\top &:= [\, \mathcal{Q}(\delta_M^1, \delta_N^1) \ \dots \ \mathcal{Q}(\delta_M^1, \delta_N^N) \,|\ \dots \ |\, \mathcal{Q}(\delta_M^M, \delta_N^1) \ \dots \ \mathcal{Q}(\delta_M^M, \delta_N^N) \,] \\ &= \ [\, \mathbf{c}_1^\top \ | \ \dots \ | \ \mathbf{c}_M^\top \,].\end{aligned}$$

Note that

$$\begin{aligned}\mathbf{c}_i^\top &= [\, \mathcal{Q}(\delta_M^i, \delta_N^1) \quad \dots \quad \mathcal{Q}(\delta_M^i, \delta_N^N) \,] \\ &= [\, \mathcal{Q}_{\delta_M^i}(\delta_N^1) \quad \dots \quad \mathcal{Q}_{\delta_M^i}(\delta_N^N) \,] \in \mathcal{L}_N.\end{aligned}$$

This implies that the index (6) can be equivalently rewritten as follows:

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{c}_f^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \tag{7}$$

where $\mathbf{c}_f \in \mathbb{R}^N$ and $\mathbf{c} \in \mathbb{R}^{NM}$.

Remark 1: As the input and state vectors take only a finite set of values, the input sequence that minimizes the cost function (7) (for any given $\mathbf{x}_0$) is the same one that minimizes

$$\tilde{J}_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = [\mathbf{c}_f + \alpha \mathbf{1}_N]^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} [\mathbf{c} + \beta \mathbf{1}_{NM}]^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t),$$

for any choice of $\alpha, \beta \in \mathbb{R}$. So, it is always possible to assume that the weight vectors $\mathbf{c}_f$ and $\mathbf{c}_i, i \in [1, M]$, are nonnegative, without affecting the optimal control solution. Therefore, in the rest of the paper, we find an input trajectory that minimizes the index cost (7) for the BCN (3), with $\mathbf{c}$ and $\mathbf{c}_f$ nonnegative vectors, starting from the initial condition $\mathbf{x}(0) = \mathbf{x}_0$.

III. Finite horizon optimal control of BCNs: problem solution

We first observe that for every choice of a family of N-dimensional real vectors $\mathbf{m}(t), t \in [0, T]$, and every state trajectory $\mathbf{x}(t), t \in [0, T]$, of the BCN, one has

$$\begin{aligned}0 &= \sum_{t=0}^{T-1} [\mathbf{m}(t+1)^\top \mathbf{x}(t+1) - \mathbf{m}(t)^\top \mathbf{x}(t)] \\ &+ \mathbf{m}(0)^\top \mathbf{x}(0) - \mathbf{m}(T)^\top \mathbf{x}(T).\end{aligned}$$

Consequently, the cost function (7) can be equivalently written as

$$\begin{aligned}J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) &= \mathbf{m}(0)^\top \mathbf{x}(0) + [\mathbf{c}_f - \mathbf{m}(T)]^\top \mathbf{x}(T) \\ &+ \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) \\ &+ \sum_{t=0}^{T-1} [\mathbf{m}(t+1)^\top \mathbf{x}(t+1) - \mathbf{m}(t)^\top \mathbf{x}(t)].\end{aligned}$$

Now, we make use of the state update equation of the BCN (3) and of the fact that, for every choice of $\mathbf{u}(t) \in \mathcal{L}_M$, one has

$$\mathbf{m}(t)^\top \mathbf{x}(t) = [\, \mathbf{m}(t)^\top \quad \mathbf{m}(t)^\top \quad \dots \quad \mathbf{m}(t)^\top \,] \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t).$$

This way we get

$$\begin{aligned} J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) &= \mathbf{m}(0)^\top \mathbf{x}(0) + [\mathbf{c}_f - \mathbf{m}(T)]^\top \mathbf{x}(T) \\ &+ \sum_{t=0}^{T-1} \big(\mathbf{c}^\top + \mathbf{m}(t+1)^\top L - \\ & \quad [\, \mathbf{m}(t)^\top \quad \dots \quad \mathbf{m}(t)^\top \,]\big) \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t). \end{aligned}$$

Now, since the values of the vectors $\mathbf{m}(t), t \in [0, T]$, do not affect the value of the index, we choose them according to the following **algorithm**:

- [Initialization] Set $\mathbf{m}(T) := \mathbf{c}_f$;
- [Recursion] For $t = T-1, T-2, \dots, 1, 0$, the jth entry of the vector $\mathbf{m}(t)$ is chosen according to the recursive rule:

$$[\mathbf{m}(t)]_j := \min_{i \in [1,M]} \Big([\mathbf{c}_i]_j + [\mathbf{m}(t+1)^\top L_i]_j \Big), \ \forall\, j \in [1, N]. \tag{8}$$

We notice that, by the previous algorithm, for every $t \in [0, T-1]$ the vector

$$\begin{aligned} \mathbf{w}(t)^\top &:= [\, \mathbf{w}_1(t)^\top \quad \mathbf{w}_2(t)^\top \quad \dots \quad \mathbf{w}_M(t)^\top \,] \\ &= [\, \mathbf{c}_1^\top \ \mid \ \dots \ \mid \ \mathbf{c}_M^\top \,] \\ &+ \mathbf{m}(t+1)^\top [\, L_1 \quad L_2 \quad \dots \quad L_M \,] \\ &- [\, \mathbf{m}(t)^\top \quad \mathbf{m}(t)^\top \quad \dots \quad \mathbf{m}(t)^\top \,] \end{aligned}$$

is nonnegative and satisfies the following condition: for every $j \in [1, N]$ there exists $i \in [1, M]$ such that $[\mathbf{w}_i(t)]_j = 0$. As a result, the index

$$\begin{aligned} J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) &= \mathbf{m}(0)^\top \mathbf{x}(0) \\ &+ \sum_{t=0}^{T-1} [\, \mathbf{w}_1(t)^\top \quad \mathbf{w}_2(t)^\top \quad \dots \quad \mathbf{w}_M(t)^\top \,] \\ & \quad \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) \end{aligned}$$

is minimized by the input sequence $\mathbf{u}(t), t \in [0, T-1]$, that is obtained according to this rule:

$$\mathbf{x}(t) = \delta_N^j \qquad \Rightarrow \qquad \mathbf{u}(t) = \delta_M^{i^*(j,t)},$$

where[1]

$$i^*(j,t) = \arg \min_{i \in [1,M]} \Big([\mathbf{c}_i]_j + [\mathbf{m}(t+1)^\top L_i]_j \Big).$$

In this way,

$$[\, \mathbf{w}_1^\top(t) \quad \dots \quad \mathbf{w}_M^\top(t) \,] \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) = 0, \ \forall\ t \in [0, T-1],$$

and by the nonnegativity of the vector $\mathbf{w}(t)$, this is the minimum possible value that this term can take.

Two straightforward consequences of the previous analysis are:

- $J_T^*(\mathbf{x}_0) := \min_{\mathbf{u}(\cdot)} J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{m}(0)^\top \mathbf{x}(0)$, where $\mathbf{m}(0)$ is obtained according to the previous algorithm.
- The optimal control input can be implemented by means of a time-varying state feedback law. Actually, the optimal input can be expressed as

$$\mathbf{u}(t) = K(t)\mathbf{x}(t),$$

where the state feedback matrix is

$$K(t) = [\, \delta_M^{i^*(1,t)} \quad \delta_M^{i^*(2,t)} \quad \dots \quad \delta_M^{i^*(N,t)} \,].$$

Remark 2: Equation (8) can be viewed as the equivalent for BCNs of the difference Riccati equation for standard discrete-time linear systems with a quadratic cost function. The updating algorithm, however, is based on a linear functional instead of a quadratic one, due to the fact that both the input and the state take only a finite number of values.

Remark 3: Bellman's Principle of Optimality [1] provides an alternative way for deriving the recursive equation (8). For any $t \in [0, T]$, let

$$J_{[t,T]}(\mathbf{x}(t), \mathbf{u}(\cdot)) := \mathbf{c}_f^\top \mathbf{x}(T) + \sum_{\tau=t}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(\tau) \ltimes \mathbf{x}(\tau)$$

denote the cost function corresponding to the interval $[t, T]$, and let $\mathbf{m}(t) \in \mathbb{R}^N$ be the row vector whose jth component is defined as follows:

$$[\mathbf{m}(t)]_j := \min_{\mathbf{u}(\cdot)} J_{[t,T]}(\delta_N^j, \mathbf{u}(\cdot)).$$

Then, it can be shown, after some manipulation, that

$$\begin{aligned} \mathbf{m}(t)^\top \mathbf{x}(t) &= \min_{\mathbf{u}(\cdot)} \Big\{ \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) \\ & \quad + \min_{\mathbf{u}(\cdot)} J_{[t+1,T]}(\mathbf{x}(t+1), \mathbf{u}(\cdot)) \Big\} \\ &= \min_{\mathbf{u}(\cdot)} \left\{ [\mathbf{c}^\top + \mathbf{m}(t+1)^\top L] \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) \right\}. \end{aligned}$$

So, the algorithm we proposed to solve the finite horizon optimal control problem is a dynamic programming algorithm, that could be alternatively derived by resorting to Bellman's Principle.

Remark 4: As far as the computational complexity of the proposed algorithm is concerned, we first consider the evaluation of the jth entry of $\mathbf{m}(t)$, $[\mathbf{m}(t)]_j$. To this goal, one needs to select the entry of $\mathbf{m}(t+1)$ corresponding to the canonical vector $[L_i]_j$ and sum such entry to $[\mathbf{c}_i]_j$. This operation must be performed for every index $i \in [1, M]$, and the smallest of these M values gives $[\mathbf{m}(t)]_j$. So, the computational burden required to evaluate $[\mathbf{m}(t)]_j$ is linear with respect to M, i.e. $O(M)$. Furthermore, this operation must be repeated for every component of $\mathbf{m}(t)$ and for every $t \in [0, T-1]$. Consequently, the computational complexity of the algorithm is $O(NMT)$.

Example 1: Consider the BCN (3) and suppose that $N = 8$, $M = 2$ and

$$\begin{aligned} L_1 &:= L \ltimes \delta_2^1 = [\, \delta_8^4 \quad \delta_8^5 \quad \delta_8^4 \quad \delta_8^5 \quad \delta_8^6 \quad \delta_8^7 \quad \delta_8^8 \quad \delta_8^7 \,], \\ L_2 &:= L \ltimes \delta_2^2 = [\, \delta_8^2 \quad \delta_8^4 \quad \delta_8^1 \quad \delta_8^7 \quad \delta_8^6 \quad \delta_8^5 \quad \delta_8^6 \quad \delta_8^6 \,]. \end{aligned}$$

We consider the problem of minimizing the cost function (7) for $T = 4$, by assuming

$$\mathbf{c}_f = [1 \quad 1 \quad 1 \quad 2 \quad 1 \quad 10 \quad 0 \quad 0]^\top, \qquad \mathbf{c} = [\, \mathbf{1}_8^\top \quad \mathbf{0}_8^\top \,]^\top,$$

[1] Note that the index that minimizes the function is not necessarily unique: so there is not necessarily a unique optimal input.

and initial condition $\mathbf{x}(0) = \delta_8^1$.

It is worth noticing that the input $\mathbf{u}(t) = \delta_2^2$ has zero cost. So, one would be tempted to just assume $\mathbf{u}(t) = \delta_2^2$ for every $t \in [0, 3]$. This way, however, $\mathbf{x}(4)$ would be equal to δ_8^6, which is the "most expensive" final state. So, we proceed according to the algorithm:

- $\mathbf{m}(4) = \mathbf{c}_f = [\,1 \quad 1 \quad 1 \quad 2 \quad 1 \quad 10 \quad 0 \quad 0\,]^\top$;
- $\mathbf{m}(3) = [\,1 \quad 2 \quad 1 \quad 0 \quad 10 \quad 1 \quad 1 \quad 1\,]^\top$ and $K(3) = [\,\delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^1 \quad \delta_2^1\,]$;
- $\mathbf{m}(2) = [\,1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 2 \quad 1 \quad 1\,]^\top$ and $K(2) = [\,\delta_2^1 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^1 \quad \delta_2^2 \quad \delta_2^2\,]$;
- $\mathbf{m}(1) = [\,0 \quad 1 \quad 1 \quad 1 \quad 2 \quad 1 \quad 2 \quad 2\,]^\top$ and $K(1) = [\,\delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2\,]$;
- $\mathbf{m}(0) = [\,1 \quad 1 \quad 0 \quad 2 \quad 1 \quad 2 \quad 1 \quad 1\,]^\top$ and $K(0) = [\,\delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2\,]$.

As a consequence,

$$J_4^*(\delta_8^1) = \min_{\mathbf{u}(\cdot)} J_4(\delta_8^1, \mathbf{u}(\cdot)) = \mathbf{m}(0)^\top \delta_8^1 = 1.$$

An optimal input sequence is

$$\mathbf{u}^*(0) = \mathbf{u}^*(1) = \mathbf{u}^*(2) = \delta_2^2, \qquad \mathbf{u}^*(3) = \delta_2^1,$$

and it corresponds to the state-trajectory $\mathbf{x}^*(0) = \delta_8^1,\ \mathbf{x}^*(1) = \delta_8^2,\ \mathbf{x}^*(2) = \delta_8^4,\ \mathbf{x}^*(3) = \delta_8^7,\ \mathbf{x}^*(4) = \delta_8^8$. ♠

IV. Extensions and special cases

A. Time-varying weight matrices

The analysis carried on in sections II and III immediately extends to the case when the cost function is time-dependent, namely

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathcal{Q}_f(\mathbf{x}(T)) + \sum_{t=0}^{T-1} \mathcal{Q}(\mathbf{u}(t), \mathbf{x}(t), t), \qquad (9)$$

where $\mathcal{Q}_f(\cdot)$ is any function defined on $\mathcal{L}_N$, and $\mathcal{Q}(\cdot,\cdot,\cdot)$ is any function defined on $\mathcal{L}_M \times \mathcal{L}_N \times [0, T-1]$. Indeed, when so, the cost function can be equivalently rewritten as

$$J(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{c}_f^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} \mathbf{c}^\top(t) \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t),$$

for $\mathbf{c}_f^\top := [\,\mathcal{Q}_f(\delta_N^1) \quad \mathcal{Q}_f(\delta_N^2) \quad \dots \quad \mathcal{Q}_f(\delta_N^N)\,]$, and

$$\begin{aligned} \mathbf{c}^\top(t) \ &:= \ [\,\mathcal{Q}(\delta_M^1, \delta_N^1, t) \quad \dots \quad \mathcal{Q}(\delta_M^1, \delta_N^N, t)\,| \\ &\quad \dots\ |\,\mathcal{Q}(\delta_M^M, \delta_N^1, t) \quad \dots \quad \mathcal{Q}(\delta_M^M, \delta_N^N, t)\,]. \end{aligned}$$

Obviously, the recursive rule (8) of the previous algorithm must be slightly modified to keep into account the fact that the cost $\mathbf{c}(t)$ is time-varying.

B. Quadratic cost function

The standard quadratic cost function

$$\begin{aligned} J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) \ &= \ \mathbf{x}(T)^\top Q_f \mathbf{x}(T) \qquad (10) \\ &+ \ \sum_{t=0}^{T-1} [\,\mathbf{x}(t)^\top \quad \mathbf{u}(t)^\top\,] \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix}, \end{aligned}$$

where Q_f, Q, S and R are matrices of suitable dimensions, can be equivalently expressed as in (7) for

$$\begin{aligned} \mathbf{c}_f^\top \ &:= \ [\,[Q_f]_{1,1} \quad [Q_f]_{2,2} \quad \dots \quad [Q_f]_{N,N}\,], \\ \mathbf{c}^\top \ &:= \ [\,\mathbf{q}^\top \quad \mathbf{q}^\top \quad \dots \quad \mathbf{q}^\top\,] \\ &+ \ [\,[R]_{1,1}\mathbf{1}_N^\top \quad [R]_{2,2}\mathbf{1}_N^\top \quad \dots \quad [R]_{M,M}\mathbf{1}_N^\top\,] \\ &+ \ 2\,[\,\mathrm{col}_1(S)^\top \quad \mathrm{col}_2(S)^\top \quad \dots \quad \mathrm{col}_M(S)^\top\,], \end{aligned}$$

and

$$\mathbf{q}^\top := [\,[Q]_{1,1} \quad [Q]_{2,2} \quad \dots \quad [Q]_{N,N}\,].$$

C. Cost function that weights only the final state

In a couple of recent papers [18], [19] the finite horizon optimal control problem for BCNs has been addressed by resorting to Pontryagin maximum principle, and by assuming as cost function a linear function of the final state:

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{c}_f^\top \mathbf{x}(T), \qquad \mathbf{c}_f \in \mathbb{R}^N. \qquad (11)$$

The problem of determining the state trajectory starting from a given $\mathbf{x}(0) = \mathbf{x}_0$ that minimizes the cost function (11) can be cast in our analysis, by simply assuming $\mathbf{c} = 0$. The algorithm becomes the following one:

- [Initialization] Set $\mathbf{m}(T) := \mathbf{c}_f$;
- [Recursion] For $t = T-1, T-2, \dots, 1, 0$, the jth entry of the vector $\mathbf{m}(t)$ is chosen according to the recursive rule:
$$[\mathbf{m}(t)]_j := \min_{i \in [1,M]} [\mathbf{m}(t+1)^\top L_i]_j,\ \forall\, j \in [1, N].$$

The minimum cost is again $J_T^*(\mathbf{x}_0) = \mathbf{m}(0)^\top \mathbf{x}_0$, and the corresponding optimal input can be expressed as a time-varying feedback from the (optimal) state trajectory. If we reverse the perspective and instead of minimizing the cost function we aim at maximizing it, as investigated in [18], [19], the algorithm requires a minor modification. Actually, at each recursion step, the goal is to select the entries of the vector $\mathbf{m}(t)$ in such a way that for every $j \in [1, N]$ there exists $i \in [1, M]$ such that $[\mathbf{m}(t+1)^\top L_i]_j - [\mathbf{m}(t)]_j = 0$, while for the other indices $\ell \neq i, \ell \in [1, M]$, $[\mathbf{m}(t+1)^\top L_\ell]_j - [\mathbf{m}(t)]_j \leq 0$. This amounts to assuming:

$$[\mathbf{m}(t)]_j := \max_{i \in [1,M]} [\mathbf{m}(t+1)^\top L_i]_j,\ \forall\, j \in [1, N].$$

Therefore also in this case the optimal input is the one that annihilates all the terms

$$\mathbf{w}(t)^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad t \in [0, T-1],$$

where

$$\mathbf{w}(t)^\top = \mathbf{m}(t+1)^\top L - [\,\mathbf{m}(t)^\top \quad \mathbf{m}(t)^\top \quad \dots \quad \mathbf{m}(t)^\top\,].$$

To illustrate this revised technique, we consider Example 8, addressed at the end of [18].

Example 2: Consider the BCN (3) and suppose that $N = 4$, $M = 4$ and

$$\begin{aligned} L_1 \ &:= \ L \ltimes \delta_4^1 = [\,\delta_4^2 \quad \delta_4^3 \quad \delta_4^3 \quad \delta_4^4\,], \\ L_2 \ &:= \ L \ltimes \delta_4^2 = [\,\delta_4^2 \quad \delta_4^1 \quad \delta_4^3 \quad \delta_4^4\,], \\ L_3 \ &:= \ L \ltimes \delta_4^3 = [\,\delta_4^1 \quad \delta_4^3 \quad \delta_4^2 \quad \delta_4^4\,], \\ L_4 \ &:= \ L \ltimes \delta_4^4 = [\,\delta_4^2 \quad \delta_4^2 \quad \delta_4^4 \quad \delta_4^3\,]. \end{aligned}$$

We address the problem of maximizing the function (11) for $T = 3$, by assuming

$$\mathbf{c}_f = [\,1 \quad 0 \quad 0 \quad 0\,]^\top,$$

and initial condition $\mathbf{x}(0) = \delta_4^4$. In order to maximize $\mathbf{c}_f^\top \mathbf{x}(3)$, we proceed according to the revised algorithm:

- $\mathbf{m}(3) \;=\; \mathbf{c}_f = [\,1 \quad 0 \quad 0 \quad 0\,]^\top$;
- $\mathbf{m}(2) \;=\; [\,1 \quad 1 \quad 0 \quad 0\,]^\top$ and $K(2) = [\,\delta_4^3 \quad \delta_4^2 \quad \delta_4^1 \quad \delta_4^1\,]$;;
- $\mathbf{m}(1) \;=\; [\,1 \quad 1 \quad 1 \quad 0\,]^\top$ and $K(1) = [\,\delta_4^2 \quad \delta_4^2 \quad \delta_4^3 \quad \delta_4^1\,]$;
- $\mathbf{m}(0) \;=\; [\,1 \quad 1 \quad 1 \quad 1\,]^\top$ and $K(0) = [\,\delta_4^2 \quad \delta_4^2 \quad \delta_4^2 \quad \delta_4^4\,]$.

As a consequence,

$$J_3^\sharp(\delta_4^4) := \max_{\mathbf{u}(\cdot)} J(\delta_4^4, \mathbf{u}(\cdot)) = \mathbf{m}(0)^\top \delta_4^4 = 1.$$

An optimal input sequence is

$$\mathbf{u}^\sharp(0) = \delta_4^4, \quad \mathbf{u}^\sharp(1) = \delta_4^3, \quad \mathbf{u}^\sharp(2) = \delta_4^2,$$

and it corresponds to the state-trajectory

$$\mathbf{x}^\sharp(0) = \delta_4^4, \quad \mathbf{x}^\sharp(1) = \delta_4^3, \quad \mathbf{x}^\sharp(2) = \delta_4^2, \quad \mathbf{x}^\sharp(3) = \delta_4^1.$$

♠

D. *Constraints on the state trajectory*

The finite horizon optimal control problem for the BCN (3) can be further enriched by introducing either constraints on the final state to be reached, or on the states and/or transitions we want to avoid. Specifically, we may be interested in imposing a specific value on the final state $\mathbf{x}(T)$. If so, we can just use the previous set-up, by assuming that $\mathbf{c}_f$ has all entries equal to $+\infty$, except for the one corresponding to the final state we want to achieve (i.e., the jth entry, if we want that $\mathbf{x}(T) = \delta_N^j$). Similarly, if we want to avoid a certain state (say δ_N^j) or a certain transition (by this meaning the use of a specific input $\mathbf{u}(t) = \delta_M^i$ corresponding to a given state $\mathbf{x}(t) = \delta_N^j$), we just need to set to $+\infty$ some entries of the cost vectors (specifically, $[\mathbf{c}_f]_j = +\infty$ and $[\mathbf{c}_i]_j = +\infty$ for all $i \in [1, M]$ in the former case, and $[\mathbf{c}_i]_j = +\infty$ for the given values of i and j in the latter case).

V. FINITE-HORIZON OPTIMAL CONTROL OF BCNS WITH PENALTY ON THE SWITCHINGS

The interpretation of a BCN as a switched Boolean system suggests a generalization of the cost function (7) we have considered up to now. Indeed, in addition to the cost on the state, weighted by the value of the input sample, we may want to penalize the switchings, namely the changes in the input value, meanwhile attributing zero cost to conservative inputs (namely to the case when $\mathbf{u}(t)$ coincides with $\mathbf{u}(t-1)$). This idea is formalized by the following cost function:

$$\begin{aligned} J_T(\mathbf{x}_0, \mathbf{u}(-1), \mathbf{u}(\cdot)) \;&=\; \mathbf{c}_f^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) \\ &+\; \sum_{t=0}^{T-1} \mathbf{p}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{u}(t-1), \end{aligned} \tag{12}$$

where $\mathbf{c}_f \in \mathbb{R}^N$, $\mathbf{c} \in \mathbb{R}^{NM}$ and $\mathbf{p} \in \mathbb{R}^{M^2}$. To penalize switchings and attribute no penalties when the input does not change, it is sufficient to choose $\mathbf{p}$ in such a way that $\mathbf{p} \ltimes \delta_M^i \ltimes \delta_M^j$ is zero if $i = j$ and positive otherwise. Clearly different switchings may be penalized in different ways.

First of all, we notice that in this set-up the initial condition is not only the state at $t = 0$, but also the input value at time $t = -1$. This suggests to tackle this problem by introducing an augmented state variable:

$$\xi(t) := \mathbf{x}(t) \ltimes \mathbf{u}(t-1),$$

with known initial condition $\xi(0) = \mathbf{x}(0) \ltimes \mathbf{u}(-1)$. As a first goal, we want to derive the one-step updating law of the variable $\xi(t)$. To this end, we need some notation. Set

$$L_{ij} := \mathrm{col}_j(L_i) = L \ltimes \delta_M^i \ltimes \delta_N^j, \qquad \forall\, i \in [1, M],\ j \in [1, N].$$

It is not difficult to verify that

$$\xi(t+1) = \tilde{L} \ltimes \mathbf{u}(t) \ltimes \xi(t), \tag{13}$$

where

$$\begin{aligned} \tilde{L} \;=\; [\,& L_{11} \ltimes \delta_M^1 \mathbf{1}_M^\top \quad \ldots \quad L_{1N} \ltimes \delta_M^1 \mathbf{1}_M^\top \;\; | \\ & L_{21} \ltimes \delta_M^2 \mathbf{1}_M^\top \quad \ldots \quad L_{2N} \ltimes \delta_M^2 \mathbf{1}_M^\top \;\; | \;\; \ldots \\ & L_{M1} \ltimes \delta_M^M \mathbf{1}_M^\top \quad \ldots \quad L_{MN} \ltimes \delta_M^M \mathbf{1}_M^\top \,] \in \mathcal{L}_{NM \times NM^2}. \end{aligned}$$

We now want to prove that also the cost function (12) can be rewritten as in (7), by referring to the state variable $\xi(t)$. Indeed, we can easily verify that

$$\begin{aligned} \mathbf{c}_f^\top \mathbf{x}(T) \;&=\; [\,[\mathbf{c}_f]_1 \mathbf{1}_M^\top \quad [\mathbf{c}_f]_2 \mathbf{1}_M^\top \quad \ldots \quad [\mathbf{c}_f]_N \mathbf{1}_M^\top\,]\, \xi(T) \\ &=\; (\mathbf{c}_f^\top \otimes \mathbf{1}_M^\top) \xi(T). \end{aligned}$$

On the other hand,

$$\begin{aligned} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) \;&= [\,[\mathbf{c}_1]_1 \mathbf{1}_M^\top \quad \ldots \quad [\mathbf{c}_1]_N \mathbf{1}_M^\top \;\; | \\ &\qquad [\mathbf{c}_2]_1 \mathbf{1}_M^\top \quad \ldots \quad [\mathbf{c}_2]_N \mathbf{1}_M^\top \;\; | \;\; \ldots \\ &\qquad [\mathbf{c}_M]_1 \mathbf{1}_M^\top \quad \ldots \quad [\mathbf{c}_M]_N \mathbf{1}_M^\top] \\ &\ltimes \;\; \mathbf{u}(t) \ltimes \xi(t) \\ &= (\mathbf{c}^\top \otimes \mathbf{1}_M^\top) \ltimes \mathbf{u}(t) \ltimes \xi(t). \end{aligned}$$

Finally, if

$$\mathbf{p}^\top = [\,\mathbf{p}_1^\top \quad \mathbf{p}_2^\top \quad \ldots \quad \mathbf{p}_M^\top\,],$$

where $\mathbf{p}_i^\top := \mathbf{p}^\top \ltimes \delta_M^i, i \in [1, M]$, then

$$\begin{aligned} & \mathbf{p}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{u}(t-1) \\ =\; & [\,\mathbf{p}_1^\top \quad \ldots \quad \mathbf{p}_1^\top \;\; | \ldots | \;\; \mathbf{p}_M^\top \quad \ldots \quad \mathbf{p}_M^\top\,] \ltimes \mathbf{u}(t) \ltimes \xi(t). \end{aligned}$$

This implies that the cost function (12) can be rewritten as

$$J_T(\xi(0), \mathbf{u}(\cdot)) = \tilde{\mathbf{c}}_f^\top \xi(T) + \sum_{t=0}^{T-1} \tilde{\mathbf{c}}^\top \ltimes \mathbf{u}(t) \ltimes \xi(t), \tag{14}$$

where

$$\begin{aligned}\tilde{\mathbf{c}}_f^\top &:= \mathbf{c}_f^\top \otimes \mathbf{1}_M^\top,\\ \tilde{\mathbf{c}}^\top &:= \mathbf{c}^\top \otimes \mathbf{1}_M^\top\\ &+ \ [\mathbf{p}_1^\top \ \ \dots \ \ \mathbf{p}_1^\top \ \ | \ \ \dots \ \ | \ \ \mathbf{p}_M^\top \ \ \dots \ \ \mathbf{p}_M^\top\,].\end{aligned}$$

Consequently, the approach developed in section III may be successfully applied to obtain the optimal solution also in this case.

Remark 5: The state variable $\xi(t)$, being a function of $\mathbf{x}(t)$ and $\mathbf{u}(t-1)$, is independent of $\mathbf{u}(t)$. Therefore $\mathbf{u}(t)$ is a free input for the BCN (13). On the other hand, it is clear that the value of $\xi(t)$ constrains the choice of $\mathbf{u}(t)$ when trying to solve the optimization problem (14). In particular, since $\xi(t)$ depends on $\mathbf{u}(t-1)$, it allows to keep into account the penalties related to switchings, when searching for an optimal solution.

VI. Infinite horizon optimal control problem

The natural extension of the previous optimal control problem to the infinite horizon case can be stated as follows:

Given the BCN (3), with initial state $\mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{L}_N$, determine an input sequence that minimizes the quadratic cost function:

$$J(\mathbf{x}_0, \mathbf{u}(\cdot)) = \sum_{t=0}^{+\infty} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad (15)$$

where $\mathbf{c} \in \mathbb{R}^{NM}$. As in the finite horizon case, we assume that the vector $\mathbf{c}$ is nonnegative.

We want to show that the only possibility of obtaining a finite value for the optimum index

$$J^*(\mathbf{x}_0) := \min_{\mathbf{u}(\cdot)} J(\mathbf{x}_0, \mathbf{u}(\cdot)),$$

is represented by the existence of a periodic state-input trajectory $(\mathbf{x}(t), \mathbf{u}(t))_{t\in\mathbb{Z}_+}$, of zero cost, that can be "reached" from $\mathbf{x}_0$. This amounts to saying that there exist $T > 0$, $\tau \geq 0$ and $\mathbf{u}(t), t \in \mathbb{Z}_+$, such that

$$(\mathbf{x}(t), \mathbf{u}(t)) = (\mathbf{x}(t+T), \mathbf{u}(t+T)), \ \forall\, t \in \mathbb{Z}_+, t \geq \tau, \quad (16)$$

and

$$\mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) = 0, \ \forall\, t \in \mathbb{Z}_+, t \geq \tau. \qquad (17)$$

Condition (17) implies, in particular, that the optimal control problem has a finite solution only if the vector $\mathbf{c}$ has some zero entry.

Proposition 1: The minimum value $J^*(\mathbf{x}_0)$ of the infinite horizon cost function (15) is finite for every choice of the initial state $\mathbf{x}_0 \in \mathcal{L}_N$ if and only if for every $\mathbf{x}_0$ there exists an input sequence that makes the resulting state-input trajectory both periodic and zero-cost starting from some time instant $\tau \geq 0$.

Proof: [Sufficiency] Let $\mathbf{x}(0) = \delta_N^J$, $j \in [1, N]$, be an arbitrary initial state, and let $\mathbf{u}^{(j)}(t), t \in \mathbb{Z}_+$, be an input sequence that makes the resulting state-input trajectory, say $(\mathbf{x}^{(j)}(t), \mathbf{u}^{(j)}(t)))$, both T-periodic and zero-cost starting from time τ, for some nonnegative integers T and τ. Then

$$J^*(\mathbf{x}_0) = \min_{\mathbf{u}(\cdot)} J(\mathbf{x}_0, \mathbf{u}(\cdot)) \leq \sum_{t=0}^{\tau-1} \mathbf{c}^\top \ltimes \mathbf{u}^{(j)}(t) \ltimes \mathbf{x}^{(j)}(t) < +\infty.$$

[Necessity] Let $\mathbf{x}_0$ be an arbitrary initial state. As the number of distinct pairs $(\mathbf{x}(t), \mathbf{u}(t)), t \geq 0$, is finite, if

$$J^*(\mathbf{x}_0) = \min_{\mathbf{u}(\cdot)} \sum_{t=0}^{+\infty} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) < +\infty,$$

then there must be some $\tau_1 > 0$ such that

$$\mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) = 0, \qquad \forall\, t \geq \tau_1.$$

On the other hand, the finite number of the distinct state-input pairs ensures that there must be two positive integers τ and T, with $\tau \geq \tau_1$, such that

$$(\mathbf{x}(\tau), \mathbf{u}(\tau)) = (\mathbf{x}(\tau+T), \mathbf{u}(\tau+T)).$$

Therefore, the state-input trajectory

$$(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) = \begin{cases}(\mathbf{x}(t), \mathbf{u}(t)), & \text{for } t \in [0, \tau+T-1];\\ (\mathbf{x}(t-T), \mathbf{u}(t-T)), & \text{for } t \in [\tau+T, +\infty),\end{cases}$$

stemming from $\tilde{\mathbf{x}}(0) = \mathbf{x}(0) = \mathbf{x}_0$, is, (at least) from $t = \tau$, periodic (of period T) with zero-cost. ■

The previous characterization essentially requires to perform two checks on the BCN: (1) to verify the existence of zero-cost periodic state-input trajectories, and (2) to check that every initial state $\mathbf{x}_0$ can "reach" (at least) one of the states belonging to any such periodic trajectory[2]. The concept of reachability has been thoroughly explored in the literature about BCNs (see, e.g., [5], [8], [20]). We say that a state $\mathbf{x}_f$ is *reachable from* $\mathbf{x}_0$ if there exist $\tau \geq 0$ and $\mathbf{u}(t), t \in \mathbb{Z}_+$, such that the state-input trajectory $(\mathbf{x}(t), \mathbf{u}(t))_{t\in\mathbb{Z}_+}$ satisfies

$$\mathbf{x}(0) = \mathbf{x}_0 \qquad \text{and} \qquad \mathbf{x}(\tau) = \mathbf{x}_f.$$

If $\mathbf{x}_0 = \delta_N^j$ and $\mathbf{x}_f = \delta_N^h$, then $\mathbf{x}_f$ is reachable from $\mathbf{x}_0$ if there exists $\tau \geq 0$ such that $[L_{tot}^\tau]_{h,j} > 0$, where

$$L_{tot} := L_1 \vee L_2 \vee \ldots \vee L_M \qquad (18)$$

is the Boolean sum of the various transition matrices L_i's.

So, we now look into these two issues. (1) *How can we check whether zero-cost periodic state-input trajectories exist?* A simple test can be performed in the following way. Let $C_i^{(0)}$ be the $N \times N$ matrix, whose columns are obtained from the cost vector $\mathbf{c}_i^\top$ according to the following rule

$$\operatorname{col}_j(C_i^{(0)}) := \begin{cases}\delta_N^j, & \text{if } [\mathbf{c}_i]_j = 0;\\ 0_N, & \text{otherwise};\end{cases} \qquad j \in [1, N].$$

It is easily seen that $L_i C_i^{(0)}$ is obtained from L_i by simply replacing with zero columns the columns corresponding to

[2] In the following, we will informally talk about zero-cost periodic state trajectory by this meaning the projection of a zero-cost periodic state-input trajectory over the state component.

state transitions (driven by the input value $\mathbf{u} = \delta_M^i$) of positive cost. Consequently,

$$L^{(0)} := (L_1 C_1^{(0)}) \vee (L_2 C_2^{(0)}) \vee \ldots \vee (L_M C_M^{(0)})$$

is the Boolean matrix representing all the state transitions that can be achieved at zero cost, provided that a suitable input is selected. In other words, $[L^{(0)}]_{h,j} = 1$ if and only if there exists $i \in [1, M]$ such that

$$\delta_N^h = L \ltimes \delta_M^i \ltimes \delta_N^j \qquad \text{and} \qquad \mathbf{c}^\top \ltimes \delta_M^i \ltimes \delta_N^j = 0.$$

So, it is clear that a zero-cost periodic state (and hence state-input) trajectory exists if and only if the digraph $\mathcal{D}(L^{(0)})$ has at least one cycle or, equivalently, $L^{(0)}$ is not nilpotent.

(2) *Is it possible from every initial state to reach at least one of the (states belonging to) periodic zero-cost state trajectories?* If $L^{(0)}$ is not nilpotent, then $(L^{(0)})^N \neq 0$. Consequently, we may introduce the set

$$H := \{h \in [1, N] : (\delta_N^h)^\top (L^{(0)})^N \neq 0\} \qquad (19)$$

of the indices of the nonzero rows in $(L^{(0)})^N$. Elementary graph theory allows us to say that $h \in H$ if and only if the state δ_N^h belongs to one of these periodic zero-cost state trajectories. From every initial state $\mathbf{x}_0 = \delta_N^j \in \mathcal{L}_N$ it is possible to reach some state $\delta_N^h, h \in H$, if and only if for every $j \in [1, N]$ the jth column of L_{tot}^N has at least one nonzero entry indexed by H.

These graph-theoretic interpretations immediately suggest a way to obtain the minimum cost $J^*(\mathbf{x}_0)$ for every $\mathbf{x}_0 = \delta_N^j \in \mathcal{L}_N$. Clearly, $J^*(\delta_N^h) = 0$ for every $h \in H$. On the other hand, for every state $\delta_N^j, j \notin H$, it is sufficient to determine the minimum cost state-input trajectory $(\mathbf{x}(t), \mathbf{u}(t))_{t \in \mathbb{Z}_+}$ starting from $\mathbf{x}(0)$ and reaching some state $\delta_N^h, h \in H$, in a finite number (at most $N - 1$) of steps.

Remark 6: It is worthwhile noticing that

$$H \subseteq H^* := \{h \in [1, N] : J^*(\delta_N^h) = 0\},$$

but the two sets do not necessarily coincide, as there may be states δ_N^j that access at zero cost some states $\delta_N^h, h \in H$, but $j \notin H$.

Under the previous assumptions, and by making use of the results regarding the finite horizon optimal control problem, we want to derive the expression of the optimal cost corresponding to any given initial condition, and to show that the optimal solution can be expressed as a static state-feedback. Let us consider the finite horizon optimal control problem

$$J_T^*(\mathbf{x}_0) := \min_{\mathbf{u}(\cdot)} \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t),$$

for the BCN (3), under the assumption that $\mathbf{x}(0) = \mathbf{x}_0$. Also, assume that the set H defined in (19) is not empty, and for every choice of $\mathbf{x}_0$ there exists at least one state $\mathbf{x}_f = \delta_N^h, h \in H$, reachable from $\mathbf{x}_0$. We want to show that, when T is sufficiently large, the finite vector sequence $\{\mathbf{m}(t)\}_{t=T,T-1,\ldots,2,1,0}$, generated by the algorithm described in section III, starting from the initial condition $\mathbf{m}(T) = 0$, converges in a finite number of steps to a nonnegative vector $\mathbf{m}^*$.

Lemma 1: The finite vector sequence $\{\mathbf{m}(t)\}_{t=T,T-1,\ldots,2,1,0}$, generated by the algorithm:

$$\begin{aligned} \mathbf{m}(T) &= 0; \\ [\mathbf{m}(t)]_j &= \min_{i \in [1,M]} [\mathbf{c}_i^\top + \mathbf{m}(t+1)^\top L_i]_j, \\ &\qquad\qquad j \in [1, N], t = T-1, T-2, \ldots, 1, 0, \end{aligned}$$

satisfies

$$0 = \mathbf{m}(T) \leq \mathbf{m}(T-1) \leq \ldots \leq \mathbf{m}(1) \leq \mathbf{m}(0). \qquad (20)$$

Moreover, for T sufficiently large, there exists $\Delta \geq 0$ such that

$$\mathbf{m}^* := \mathbf{m}(T - \Delta) = \mathbf{m}(T - \tau), \qquad \forall\, \tau \in [\Delta, T]. \qquad (21)$$

Proof: From the analysis of the finite horizon optimal control problem, we deduce that, for every choice of the initial state $\mathbf{x}_0 = \delta_N^j, j \in [1, N]$, and every $\tau \geq 1$,

$$\begin{aligned} \mathbf{m}(T-\tau)^\top \delta_N^j &= \min_{\mathbf{u}(\cdot)} \sum_{t=T-\tau}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \\ &\qquad (\text{with } \ \mathbf{x}(T-\tau) = \delta_N^j) \\ &= \min_{\mathbf{u}(\cdot)} \sum_{t=0}^{\tau-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \\ &\qquad (\text{with } \ \mathbf{x}(0) = \delta_N^j) \\ &\leq \min_{\mathbf{u}(\cdot)} \sum_{t=0}^{\tau} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \\ &\qquad (\text{with } \ \mathbf{x}(0) = \delta_N^j) \\ &= \min_{\mathbf{u}(\cdot)} \sum_{t=T-\tau-1}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \\ &\qquad (\text{with } \ \mathbf{x}(T-\tau-1) = \delta_N^j) \\ &= \mathbf{m}(T-\tau-1)^\top \delta_N^j. \end{aligned}$$

The arbitrariety of j ensures that $\mathbf{m}(T-\tau) \leq \mathbf{m}(T-\tau-1)$.

We now prove that the sequence is upper bounded. Actually, let d_j be the total cost of a state-input trajectory that drives the system state from δ_N^j to some state $\delta_N^h, h \in H$ (and does not pass through the same state twice). Clearly, for every $\tau \geq N$,

$$[\mathbf{m}(T-\tau)]_j = \mathbf{m}(T-\tau)^\top \delta_N^j = \min_{\mathbf{u}(\cdot)} \sum_{t=0}^{\tau-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) \leq d_j.$$

Consequently,

$$\mathbf{m}(T-\tau)^\top \leq [\, d_1 \quad d_2 \quad \ldots \quad d_N \,].$$

Finally we prove that this upper-bounded, monotonically increasing sequence converges to its limit value in a finite number of steps. Set $c_{\min} := \min\{[\mathbf{c}_i]_j > 0 : i \in [1, M], j \in [1, N]\}$. By the interpretation of $[\mathbf{m}(T-\Delta)]_j$ as the minimal cost over an interval of length Δ (specifically $[0, \Delta - 1]$) starting from the state δ_N^j, it follows that the sequence

$$0 = [\mathbf{m}(T)]_j \leq [\mathbf{m}(T-1)]_j \leq \ldots \leq [\mathbf{m}(1)]_j \leq [\mathbf{m}(0)]_j$$

at each time step, namely when moving from $T-\tau$ to $T-\tau-1$, either remains constant or it increases of at least $c_{\min}$. On the other hand, if the sequence would remain constant for $N+1$ consecutive time instants, say

$$[\mathbf{m}(T-\tau)]_j = [\mathbf{m}(T-\tau-1)]_j = \ldots = [\mathbf{m}(T-\tau-N)]_j$$

this would mean that the optimal cost starting from δ_N^j would not change when moving the final time instant from τ to $\tau+N$. This implies that the optimal state trajectory has entered (at least at time $\tau+N-1$) a zero-cost loop. But then $[\mathbf{m}(T-t)]_j = [\mathbf{m}(T-\tau)]_j$ for all $t \in [\tau, T]$. So, we have shown that every jth entry of $\mathbf{m}(T-\tau)$ is upper bounded and not decreasing as τ increases, it cannot remain constant for more than N consecutive time instants, and every time it increases it increases of at least $c_{\min}$. This ensures that it reaches its maximum value in a finite number of steps. Since this is true for every $j \in [1, N]$, the proof is completed. ■

As an immediate consequence of the previous lemma, we can claim what follows:

Theorem 1: Assume that the set H defined in (19) is not empty, and for every choice of $\mathbf{x}_0$ there exists at least one state $\mathbf{x}_f = \delta_N^h, h \in H$, reachable from $\mathbf{x}_0$. Then

1) there exists $\bar{T} \geq 0$ such that, for every $\mathbf{x}_0$,

$$J_T^*(\mathbf{x}_0) = J_{\bar{T}}^*(\mathbf{x}_0) = (\mathbf{m}^*)^\top \mathbf{x}_0, \qquad \forall\, T \geq \bar{T},$$

and therefore

$$J^*(\mathbf{x}_0) = \min_{\mathbf{u}(\cdot)} \sum_{t=0}^{+\infty} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) = (\mathbf{m}^*)^\top \mathbf{x}_0.$$

2) $\mathbf{m}^*$ is obtained through the algorithm of section III, by assuming $\mathbf{c}_f = 0$, and is a fixed point of the algorithm, namely a solution of the family of equations:

$$[\mathbf{m}^*]_j = \min_{i\in[1,M]} [\mathbf{c}_i^\top + (\mathbf{m}^*)^\top L_i]_j, \qquad j \in [1, N], \quad (22)$$

that represent the equivalent, for BCNs, of the algebraic Riccati equation for linear systems.

3) Upon defining

$$i^*(j) := \arg \min_{i\in[1,M]} [\mathbf{c}_i^\top + (\mathbf{m}^*)^\top L_i]_j,$$

the optimal control input can be implemented by means of the static state-feedback law:

$$\mathbf{u}(t) = K\mathbf{x}(t),$$

where the (not necessarily unique) feedback matrix is

$$K = [\, \delta_M^{i^*(1)} \quad \delta_M^{i^*(2)} \quad \ldots \quad \delta_M^{i^*(N)} \,].$$

Remark 7: As far as the computational complexity of the algorithm proposed in Theorem 1 is concerned, we have already remarked that the evaluation of the vector sequence $\mathbf{m}(t), t \in [0, T-1]$, has complexity $O(NMT)$. On the other hand, the graph theoretic interpretation of the cost vectors ensures that when the limit $\mathbf{m}^*$ is finite, it coincides with the vector $\mathbf{m}(0)$ obtained for the finite horizon optimal control problem in $[0, T]$, provided that $T \geq N$. Therefore, the complexity of Theorem 1 is $O(N^2 M)$.

The family of equations (22) admits an infinite number of nonnegative solutions. We want to prove that $\mathbf{m}^*$ is the smallest among them.

Proposition 2: Assume that the set H defined in (19) is not empty, and for every choice of $\mathbf{x}_0$ there exists at least one state $\mathbf{x}_f = \delta_N^h, h \in H$, reachable from $\mathbf{x}_0$. If $\mathbf{m}$ is a nonnegative solution of the family of equations (22), then $\mathbf{m} \geq \mathbf{m}^*$.

Proof: Consider, again, the finite horizon optimal control problem

$$J_T^*(\mathbf{x}_0) = \min_{\mathbf{u}(\cdot)} \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t),$$

for the BCN (3), under the assumption that $\mathbf{x}(0) = \mathbf{x}_0$. By Theorem 1, we know that for every $T \geq \bar{T}$, $J_T^*(\mathbf{x}_0) = (\mathbf{m}^*)^\top \mathbf{x}_0$. On the other hand,

$$\begin{aligned} J_T^*(\mathbf{x}_0) = {} & \min_{\mathbf{u}(\cdot)} \Big(\mathbf{m}(0)^\top \mathbf{x}(0) - \mathbf{m}(T)^\top \mathbf{x}(T) \qquad (23) \\ & + \sum_{t=0}^{T-1} \big(\mathbf{c}^\top + \mathbf{m}(t+1)^\top L - [\, \mathbf{m}(t)^\top \quad \ldots \quad \mathbf{m}(t)^\top \,] \big) \\ & \ltimes \ \mathbf{u}(t) \ltimes \mathbf{x}(t) \Big), \end{aligned}$$

for every choice of the vector sequence $\mathbf{m}(t), t \in [0, T]$. This applies in particular if we assume $\mathbf{m}(t) = \mathbf{m}, \forall\ t \in [0, T]$. When so, the choice

$$\mathbf{u}(t) = \delta_N^{i(j,t)}, \quad \forall\ t \in [0, T-1],$$

where

$$i(j,t) = \arg \min_{i\in[1,M]} \Big([\mathbf{c}_i]_j + [\mathbf{m}^\top L_i]_j \Big),$$

ensures that all the terms in (23) (for $\mathbf{m}(t) = \mathbf{m}(t+1) = \mathbf{m}$) become zero, and hence

$$\begin{aligned} (\mathbf{m}^*)^\top \mathbf{x}_0 &= \min_{\mathbf{u}(\cdot)} \Big(\mathbf{m}^\top \mathbf{x}(0) - \mathbf{m}^\top \mathbf{x}(T) \\ &+ \sum_{t=0}^{T-1} \big(\mathbf{c}^\top + \mathbf{m}^\top L - [\, \mathbf{m}^\top \quad \ldots \quad \mathbf{m}^\top \,] \big) \\ &\ltimes \ \mathbf{u}(t) \ltimes \mathbf{x}(t) \Big) \\ &\leq \mathbf{m}^\top \mathbf{x}(0) - \mathbf{m}^\top \mathbf{x}(T) \leq \mathbf{m}^\top \mathbf{x}_0. \end{aligned}$$

The arbitrariety of $\mathbf{x}_0$ ensures that $\mathbf{m}^* \leq \mathbf{m}$. ■

The previous results point out two ways to obtain the vector $\mathbf{m}^*$: (1) as the limit solution of the algorithm given in section III, initialized with the zero vector, or (2) as the smallest nonnegative solution of (22). As far as the first method is concerned, the following example shows that even if the algorithm always converges to $\mathbf{m}^*$ in a finite number of steps, say Δ, however there is no upper bound on Δ, as it depends on the specific choice of the cost vector $\mathbf{c}$. This means that the algorithm may converge to $\mathbf{m}^*$ very slowly.

Example 3: Consider the BCN (3) and suppose that $N = 4$, $M = 2$, and

$$\begin{aligned} L_1 &:= L \ltimes \delta_2^1 = [\, \delta_4^2 \quad \delta_4^3 \quad \delta_4^2 \quad \delta_4^4 \,], \\ L_2 &:= L \ltimes \delta_2^2 = [\, \delta_4^1 \quad \delta_4^4 \quad \delta_4^2 \quad \delta_4^1 \,]. \end{aligned}$$

Assume as cost function the one associated with the vector

$$\mathbf{c}^\top \quad = \quad [1 \quad \varepsilon \quad \varepsilon \quad 0 \quad 1 \quad 1 \quad 1 \quad 2],$$

where $\varepsilon > 0$ is arbitrarily small, and let $\mathbf{x}(0) = \delta_4^1$ be the initial condition. The BCN can be represented by the following digraph (see Fig. 1), obtained by overlapping the two digraphs $\mathcal{D}(L_1)$ and $\mathcal{D}(L_2)$. (Blue) continuous arcs belong to $\mathcal{D}(L_1)$, while (red) dashed arcs belong to $\mathcal{D}(L_2)$. For each vertex j (corresponding to δ_4^j), there are two outgoing arcs: each of them labelled by the input and cost values, i.e. by the pair $(\mathbf{u}, \mathbf{c}^\top \ltimes \mathbf{u} \ltimes \delta_4^j)$.

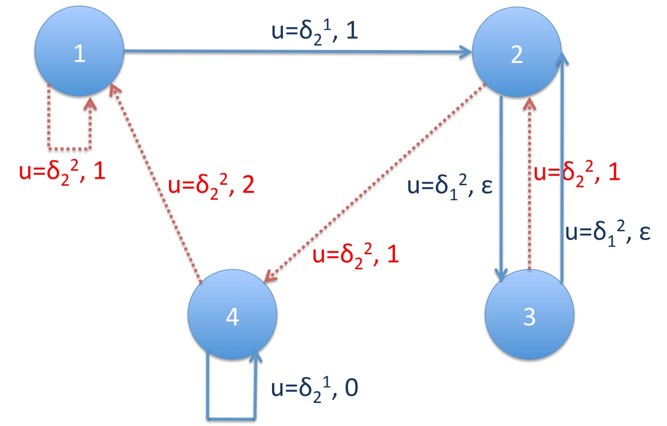

Fig. 1: Digraph corresponding to the BCN of Example 3

Set $\Delta := \min\{t \in \mathbb{Z}_+ : (t-1)\varepsilon \geq 1\} = \lceil \frac{1}{\varepsilon} \rceil + 1$. It is a matter of simple computation to prove (or to derive from the digraph) that

$$[\mathbf{m}(T-t)]_1 = \begin{cases} 1 + (t-1)\varepsilon, & \forall\, t \in [1, \Delta); \\ 2, & \forall\, t \in [\Delta, +\infty). \end{cases}$$

Accordingly, as far as the optimal control problem (with no final cost) is posed over the finite horizon $[0, T], T < \Delta$, the optimal input is constant and equal to $\mathbf{u}^*(t) = \delta_2^1$, while for $T \geq \Delta$ (in particular, for the infinite horizon optimal control problem) the optimal input is

$$\mathbf{u}^*(t) = \begin{cases} \delta_2^1, & \text{for } t = 0; \\ \delta_2^2, & \text{for } t = 1; \\ \delta_2^1 & \text{for } t \geq 2. \end{cases}$$

♠

As far as the second method is concerned, there is no need for computing all the nonnegative solutions of (22) in order to identify $\mathbf{m}^*$ among them. Indeed, we may notice that for every $h \in H$, $[\mathbf{m}^*]_h = 0$. This trivially follows from the fact that for every initial state $\delta_N^h, h \in H$, the zero-cost periodic state-input trajectory departing from that state is surely optimal, and hence $0 = J^*(\delta_N^h) = (\mathbf{m}^*)^\top \delta_N^h$. On the other hand, the following proposition shows that $\mathbf{m}^*$ is the only solution of (22) taking zero values on H. So, we can easily evaluate $\mathbf{m}^*$ from (22), by preliminarily determining H through (19) and by setting to zero all the entries of $\mathbf{m}^*$ indexed by H.

Proposition 3: Assume that the set H defined in (19) is not empty, and for every choice of $\mathbf{x}_0$ there exists at least one state $\mathbf{x}_f = \delta_N^h, h \in H$, reachable from $\mathbf{x}_0$. If $\mathbf{m}$ is a solution of the family of equations (22) and $[\mathbf{m}]_h = 0$ for every $h \in H$, then $\mathbf{m} = \mathbf{m}^*$.

Proof: We already know that there exists a solution of (22), namely $\mathbf{m}^*$, whose entries indexed by H are zero. So, it is sufficient to show that the entries $[\mathbf{m}]_j, j \notin H$, of any solution $\mathbf{m}$ of (22) are uniquely determined by the values of the entries $[\mathbf{m}]_h, h \in H$. Let H_k denote the set of indices $j \in [1, N]$ of the states δ_N^j for which the (state/input) path of minimal cost leading to some state $\delta_N^h, h \in H$, has length k. We observe that for every $j \in H_1$ there exists $q \in [1, M]$ such that $L \ltimes \delta_M^q \ltimes \delta_N^j = \delta_N^h$ for some $h \in H$, and

$$[\mathbf{m}]_j = \left[\mathbf{c}_q^\top + \mathbf{m}^\top L_q\right]_j = [\mathbf{c}_q]_j + [\mathbf{m}]_h = [\mathbf{c}_q]_j\,.$$

So, for every $j \in H_1$, the jth entry of $\mathbf{m}$ is uniquely determined.

Now we observe that H_2 can be seen as the set of indices $j \in [1, N]$ of the states δ_N^j for which the (state/input) path of minimal cost leading to some state $\delta_N^h, h \in H_1$, has length 1. So, by recursively applying the previous reasoning, we show that, for every $j \in H_2$, the jth entry of $\mathbf{m}$ is uniquely determined, and so on. ■

To conclude, it is worthwhile to briefly comment on the various nonnegative solutions of the equations (22). It is possible to prove that if we denote by $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$ the disjoint (not necessarily elementary) zero-cost cycles in $\mathcal{D}(L^{(0)})$ (so that $H = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \ldots \cup \mathcal{C}_k$), then

$$[\mathbf{m}]_h = [\mathbf{m}]_\ell, \qquad \forall\ h, \ell \in \mathcal{C}_r, \quad \forall\ r \in [1, k].$$

So, if the values of these entries, corresponding to the states belonging to the various zero-cost states, have been assigned as

$$[\mathbf{m}]_h = \alpha_r, \qquad \forall\ h \in \mathcal{C}_r, \quad r \in [1, k],$$

all the other entries are uniquely determined. However, the nonnegative values of $\alpha_1, \alpha_2, \dots, \alpha_k$ are not completely arbitrary. Indeed, they are constrained by the fact that for each $h \in \mathcal{C}_r$ and $q \in \mathcal{C}_s, r \neq s$, if δ_N^q can be reached from δ_N^h, then it must be

$$\alpha_r = [\mathbf{m}]_h \leq w(h, q) + [\mathbf{m}]_q = w(h, q) + \alpha_s,$$

where $w(h, q)$ is the cost of the minimum cost path from δ_N^h to δ_N^q.

Example 4: Consider the BCN (3) and suppose that $N = 6$, $M = 2$ and

$$\begin{aligned} L_1 \quad &:= \quad L \ltimes \delta_2^1 = [\,\delta_6^3 \quad \delta_6^1 \quad \delta_6^4 \quad \delta_6^5 \quad \delta_6^6 \quad \delta_6^6\,], \\ L_2 \quad &:= \quad L \ltimes \delta_2^2 = [\,\delta_6^3 \quad \delta_6^3 \quad \delta_6^3 \quad \delta_6^3 \quad \delta_6^4 \quad \delta_6^2\,]. \end{aligned}$$

We consider the problem of minimizing the cost function (15), by assuming

$$\mathbf{c}^\top \quad = \quad [1 \quad 1 \quad 0 \quad 5 \quad 2 \quad 0 \quad 7 \quad 2 \quad 0 \quad 0 \quad 2 \quad 4],$$

and arbitrary initial condition $\mathbf{x}(0)$. The BCN can be represented by the digraph illustrated in Fig. 2.

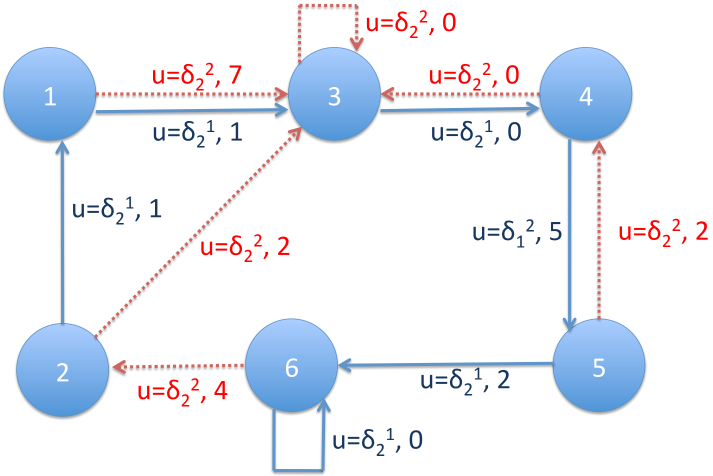

Fig. 2: Digraph corresponding to the BCN of Example 4

It is a matter of simple computation to prove (or to derive from the digraph) that

$$\mathbf{m}^* = [1 \quad 2 \quad 0 \quad 0 \quad 2 \quad 0]^\top$$

and

$$K = [\,\delta_2^1 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^2 \quad \delta_2^1 \quad \delta_2^1\,]\,.$$

Also, it is easy to verify that the disjoint zero-cost cycles are $\mathcal{C}_1 : 3 \to 3 \to 4 \to 3$ and $\mathcal{C}_2 : 6 \to 6$. So, the solutions of (22) are those and those only that can be parametrized as follows:

$$\mathbf{m} = [\,1+\alpha_1 \quad 2+\alpha_1 \quad \alpha_1 \quad \alpha_1 \quad 2+\min\{\alpha_1,\alpha_2\} \quad \alpha_2\,]^\top,$$

where α_1, α_2 are nonnegative parameters, satisfying either one of the following constraints:

$$\alpha_1 \le \alpha_2 \le \alpha_1 + 6 \qquad \text{or} \qquad \alpha_2 \le \alpha_1 \le \alpha_2 + 7.$$

Clearly, $\mathbf{m}^*$ is the smallest solution, corresponding to $\alpha_1 = \alpha_2 = 0$. ♠

VII. Average cost minimization

In [30] and [8], the problem of finding the input sequence that maximizes, on the infinite horizon, some average payoff that weights both the state and the input at every time $t \in \mathbb{Z}_+$, has been investigated. We want to show that the analysis carried on in section III for the finite-horizon optimal control problem can be used to provide a solution also to that problem. To be consistent with the previous analysis, we will search for the minimum average cost instead of the maximum payoff[3], and hence address the following problem:

Given the BCN (3), with initial state $\mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{L}_N$, determine an input sequence that minimizes the average cost function:

$$J_a(\mathbf{x}_0, \mathbf{u}(\cdot)) = \lim_{T\to+\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad (24)$$

where $\mathbf{c} \in \mathbb{R}^{NM}$ is a nonnegative vector.

We first observe that the input sequence that solves the finite horizon optimum problem

$$J_T^*(\mathbf{x}_0) = \min_{\mathbf{u}(\cdot)} \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad (25)$$

for the BCN (3), under the assumption that $\mathbf{x}(0) = \mathbf{x}_0$, also solves the finite horizon optimum problem

$$\min_{\mathbf{u}(\cdot)} \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t). \qquad (26)$$

Moreover, if we denote by $\tilde{\mathbf{m}}_T(0)$ the vector $\mathbf{m}(0)$ obtained from the usual algorithm, applied over the time interval $[0, T]$ and initialized by $\mathbf{m}(T) = 0$, we know that

$$\min_{\mathbf{u}(\cdot)} \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) = \frac{1}{T} \tilde{\mathbf{m}}_T(0)^\top \mathbf{x}_0.$$

So, it is immediately seen that

$$J_a^*(\mathbf{x}_0) := \min_{\mathbf{u}(\cdot)} J_a(\mathbf{x}_0, \mathbf{u}(\cdot)) = \lim_{T\to+\infty} \frac{1}{T} \tilde{\mathbf{m}}_T(0)^\top \mathbf{x}_0 = \mathbf{m}_a^\top \mathbf{x}_0,$$

where

$$\mathbf{m}_a := \lim_{T\to+\infty} \frac{1}{T} \tilde{\mathbf{m}}_T(0). \qquad (27)$$

It is worth noticing that the limit always exists, independently of the structure of the digraph corresponding to the BCN (in particular, of the existence of zero-cost periodic state-input trajectories reachable from the various initial states), since $\mathbf{m}_a \le \max_{i\in[1,M], j\in[1,N]} [\mathbf{c}_i]_j \mathbf{1}_N$. Consequently, the optimal average cost problem has always a finite solution. Moreover, the identity (27) together with the recursive algorithm given in section III provide a simple way to obtain the vector $\mathbf{m}_a$ of the minimal average costs.

As far as the optimal input is concerned, as previously clarified, the input sequence that solves the finite-horizon optimum problem (25) also solves (26). So, it is clear that the input sequence that minimizes the average cost (24) can be determined by suitably adjusting the algorithm presented in the previous section. Also in this case it is possible to prove that the optimal input becomes periodic after a finite number of steps and it can be expressed as a static state-feedback.

The approach to the problem solution we just described is purely numerical and it does not require any insight into the digraph associated with the BCN. On the other hand, the solution provided by D. Cheng and co-authors is entirely based on the graph theoretic interpretation of the optimal solution. Indeed, as proved in [30] and [8], for every initial condition $\mathbf{x}_0 \in \mathcal{L}_N$ the minimum average cost is achieved by eventually leading the state-input evolution $(\mathbf{x}(t), \mathbf{u}(t))_{t\in\mathbb{Z}_+}$ to (one of) the periodic state-input trajectory of minimum average cost reachable from $\mathbf{x}_0$. In detail, for every $\mathbf{x}_0 \in \mathcal{L}_N$ we consider all the elementary cycles $\mathcal{C}_i$ in $\mathcal{D}(L_{tot})$ that are reachable from $\mathbf{x}_0$ (by this meaning that one, and hence all, of the states belonging to $\mathcal{C}_i$ are reachable from $\mathbf{x}_0$). If $\mathcal{C}_i : j_1 \to j_2 \to \ldots \to j_d \to j_1$ and the transition from the vertex j_h to the vertex j_{h+1} (with $j_{d+1} := j_1$) is due to an arc appearing

[3] The adaptation to the maximum payoff case of the present analysis can be obtained by replacing the "min" function with the "max" in the algorithm that generates the vector sequence $\mathbf{m}(t), t \in \mathbb{Z}_+$, similarly to what we did in section IV.C, for the cost function that weights only the final state.

(also) in $\mathcal{D}(L_{i_h})$ (equivalently, $\delta_N^{j_{h+1}} = L \ltimes \delta_M^{i_h} \ltimes \delta_N^{j_h}$), then the average cost of $\mathcal{C}_i$ is

$$C_a(\mathcal{C}_i) = \frac{1}{d}\sum_{h=1}^{d} \mathbf{c}^\top \ltimes \delta_M^{i_h} \ltimes \delta_N^{j_h}. \qquad (28)$$

Therefore

$$J_a^*(\mathbf{x}_0) = \min\{C_a(\mathcal{C}_i) : \mathcal{C}_i \text{ is reachable from } \mathbf{x}_0\}.$$

In [30] and [8], it is also proved that a logical matrix G^* can be found such that the optimal state and input trajectories update according to the following equations:

$$\begin{cases} \mathbf{x}^*(t+1) &= L \ltimes \mathbf{u}^*(t) \ltimes \mathbf{x}^*(t) \\ \mathbf{u}^*(t+1) &= G^* \ltimes \mathbf{u}^*(t) \ltimes \mathbf{x}^*(t). \end{cases}$$

This method is very elegant and has an appealing graph theoretic interpretation, but it also requires to evaluate all the state-input cycles of the digraph associated with the BCN and their cost.

To conclude, we want to observe that once we take for granted (28) (and hence also the knowledge of the minimum cost cycles reachable from the various states), we can suitably adjust the analysis developed in the infinite horizon case in section VI, to quickly find $\mathbf{m}_a$ and the state-feedback matrix that determines the optimal input solution. To this end it is sufficient to solve the infinite horizon minimum cost problem, by suitably modifying the vector $\mathbf{c}$. In detail, we can proceed as follows:

- First of all, for every $\mathbf{x}_0$, we select the cycle of minimum average cost that can be reached from $\mathbf{x}_0$. If it is not unique, we choose the one that can be reached from $\mathbf{x}_0$ through the minimum cost path. We denote such cycle by $\mathcal{C}_i(\mathbf{x}_0) : j_1 \to j_2 \to \ldots \to j_d \to j_1$.
- If the transition of minimum cost from the vertex j_h to the vertex j_{h+1} (by assuming, again, $j_{d+1} = j_1$) is due to $\mathbf{u} = \delta_M^{i_h}$, then we replace $[\mathbf{c}_{i_h}]_{j_h}$ with the zero entry in the cost vector $\mathbf{c}$.
- After this modification of the cost vector, the optimal input trajectory that solves the original minimum average cost problem is the same one that solves the infinite horizon optimal control problem for the new vector $\mathbf{c}$. Accordingly, we can introduce the set H of all states that belong to zero-cost cycles, impose $[\mathbf{m}^*]_h = 0, \forall\, h \in H$, and determine the remaining entries of the vector $\mathbf{m}^*$ through (22). The corresponding state-feedback matrix K will be determined from the algorithm (22), as described in Theorem 1.

References

[1] R.E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ. ,1957. Republished 2003: Dover, ISBN 0-486-42809-5.

[2] R. A. Brualdi and H. J. Ryser. *Combinatorial matrix theory*. Cambridge Univ. Press, 1991.

[3] D. Cheng. Input-state approach to Boolean Networks. *IEEE Trans. Neural Networks*, 20, (3):512 – 521, 2009.

[4] D. Cheng and J.B. Liu. Stabilization of Boolean control networks. In *Proc. of the Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 5269–5274, Shanghai, China, 2009.

[5] D. Cheng and H. Qi. Controllability and observability of Boolean control networks. *Automatica*, 45:1659–1667, 2009.

[6] D. Cheng and H. Qi. Linear representation of dynamics of Boolean Networks. *IEEE Trans. Automatic Control*, 55, (10):2251 – 2258, 2010.

[7] D. Cheng and H. Qi. State-space analysis of Boolean Networks. *IEEE Trans. Neural Networks*, 21, (4):584 – 594, 2010.

[8] D. Cheng, H. Qi, and Z. Li. *Analysis and control of Boolean networks*. Springer-Verlag, London, 2011.

[9] W.K. Ching, S.Q. Zhang, Y. Jiao, T. Akutsu, N.K. Tsing, and A.S. Wong. Optimal control policy for probabilistic Boolean networks with hard constraints. *IET Syst Biol.*, 3 (2):90–99, 2009.

[10] B. Faryabi, G. Vahedi, J.-F. Chamberland, A. Datta, and E.R. Dougherty. Optimal constrained stationary intervention in gene regulatory networks. *EURASIP J. Bioinformatics and Systems Biology*, Article ID 620767:10 pages doi:10.1155/2008/620767, 2008.

[11] A. Faure, A. Naldi, C. Chaouiya, and D. Thieffry. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22:e124–e131, 2006.

[12] E. Fornasini and M. E. Valcher. Observability, reconstructibility and state observers of Boolean control networks. *IEEE Tran. Aut. Contr.*, 58 (6):1390 – 1401, 2013.

[13] E. Fornasini and M. E. Valcher. On the periodic trajectories of Boolean Control Networks. *Automatica*, 49:1506–1509, 2013.

[14] E. Fornasini and M.E. Valcher. Observability and reconstructibility of Boolean control networks. In *Proc. of the 51st Conference on Decision and Control (CDC2012)*, pages 2574–2580, Maui, Hawaii, 2012.

[15] D. G. Green, T. G. Leishman, and S. Sadedin. The emergence of social consensus in Boolean networks. In *Proc. IEEE Symp. Artificial Life (ALIFE07)*, pages 402–408, Honolulu, HI, 2007.

[16] G. Hochma, M. Margaliot, E. Fornasini, and M.E. Valcher. Symbolic dynamics of boolean control networks. *Automatica*, 49 (8):2525–2530, 2013.

[17] S.A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theoretical Biology*, 22:437467, 1969.

[18] D. Laschov and Margaliot M. A pontryagin maximum principle for multi-input boolean control networks. In . Kaslik and S. Sivasundaram, editors, *Recent Advances in Dynamics and Control of Neural Networks*. to appear, 2013.

[19] D. Laschov and M. Margaliot. A maximum principle for single-input Boolean Control Networks. *IEEE Trans. Automatic Control*, 56, no. 4:913–917, 2011.

[20] D. Laschov and M. Margaliot. Controllability of Boolean control networks via the Perron-Frobenius theory. *Automatica*, 48:1218–1223, 2012.

[21] H. Li. Global stability and controllability of switched Boolean networks. In *Proc. of the 31th Chinese Control Conference*, pages 82–88, Hefei, China, 2012.

[22] H. Li and Y. Wang. Boolean derivative calculation with application to fault detection of combinational circuits via the semi-tensor product method. *Automatica*, 48, (4):688–693, 2012.

[23] H. Li and Y. Wang. On reachability and controllability of switched Boolean control networks. *Automatica*, 48 (11):29172922, 2012.

[24] H. Li and Y. Wang. Consistent stabilizability of switched Boolean networks. *Neural Networks*, 46:183189, 2013.

[25] Q. Liu. Optimal finite horizon control in gene regulatory networks. *Eur. Phys. J. B*, 86: 245:1–5,DOI: 10.1140/epjb/e2013–30746–7, 2013.

[26] Q. Liu, X. Guo, and T. Zhou. Optimal control for probabilistic Boolean networks. *IET Syst Biol.*, 4 (2):99–107, 2010.

[27] R. Pal, A. Datta, and E.R. Dougherty. Optimal infinite-horizon control for probabilistic Boolean Networks. *IEEE Trans. Sign. Proc.*, 54 (6):2375–2397, 2006.

[28] I. Shmulevich, E.R. Dougherty, S. Kim, and W. Zhang. From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proc. IEEE*, 90, no. 11:1778–1792, 2002.

[29] C. Yang, C. Wai-Ki, T. Nam-Kiu, and L. Ho-Yin. On finite-horizon control of genetic regulatory networks with multiple hard-constraints. *BMC Systems Biology*, 4 (Suppl. 2):S14:1–7,DOI: 10.1186/1752–0509–4–S2–S14, 2010.

[30] Y. Zhao, Z. Li, and D. Cheng. Optimal control of logical control networks. *IEEE Trans. Automatic Control*, 56, (8):1766–1776, 2011.

[31] Q. Zhu and G. Xie. Finite-horizon optimal control of discrete-time switched linear systems. *Mathematical Problems in Engineering*, Article ID 483568:doi:10.1155/2012/483568, 2012.