Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A Matrix Expression of Infinite Horizon Optimal Control Problem for Stochastic Logical Dynamical Systems

Yuhu Wu [*,**] **Tielong Shen** [*]

[*] *Department of Electrical Engineering, Sophia University, Tokyo, 102-8554, Japan (e-mail: tetu-sin@sophia.ac.jp)*
[**] *Department of Mathematics, Harbin University of Science and Technology, Harbin 150080, PR China (e-mail: wuyuhu51@gmail.com)*

Abstract: The stochastic logical control dynamical system with finite state is considered. After giving two equivalent descriptions of stochastic logical dynamical system: in term of discrete-time evolution equation and in term of Markov process, the infinite horizon optimization problem is presented in an algebraic form. Based on semi-tensor product of matrix and the increasing dimensional technique, we establish a succinct matrix expression of dynamic programming and Bellman's equation for the optimal control problem.

Keywords: Logical control, Infinite horizon optimal control, Stochastic logical dynamical systems, Semi-tensor product of matrices

1. INTRODUCTION

In dynamical systems theory so-called logical dynamical system has been widely investigated for the last two decades in which the state variable is defined in a logic domain (or Boolean domain) Trumper [2004]. Usually, the logic domain consist of a finite or an infinite logic elements. For such kind of systems in which the state of system might be represented as a logic variable, analysis and synthesis of the system in a logical dynamical system framework usually leads to simple logical control law. So, the logical-based control becomes a hot topic in control community. Stability and stabilization of logical dynamics can be found in Akutsu et al. [2007], Li & Sun [2011]. Regarding the stochastic characteristic, logical dynamical systems with stochastic properties have been addressed in several literatures, Doberkat [2011], Zhang et al. [2011]. Finite or infinite horizon optimal control problems for stochastic logical (Boolean) systems have been investigated in Karaman et al. [2008] and Oal & Datta [2006]. The application of stochastic logical control have enriched in various fields, including genetic regulatory networks [Faryabi et al. , 2007] and internal combustion engines control [Wu & Shen, 2013]. In this paper, the infinite horizon optimal control problem for the stochastic logical dynamical system with finite state is considered. The purpose of this work is to give a succinct matrix expression of infinite horizon optimal control algorithm for the stochastic logical dynamical systems, via semi-tensor product and the increasing dimensional technique.

The rest of this paper is organized as follows. Section 2 is the problem formulation, which introduces some notations and concepts of infinite horizon optimal control for stochastic logical dynamical system. In section 3, after giving a brief survey on semi-tensor product, we give the matrix expression of the optimization problem under Markov process description of the stochastic logical dynamical system. We convert the dynamic programming algorithm and Bellman's equation for this infinite horizon optimal control problem into simple algebraic form, in section 4. Section 5 is a brief conclusion.

2. PROBLEM FORMULATION

Assume the logic state space consist of finite element, denoted by $S = \{x^1, x^2, \cdots, x^s\}$. We also assume the control space U also consist of finite elements, $U = \{u^1, u^2, \cdots, u^r\}$.

The most usage way to represent stochastic dynamics with control is described by a stationary discrete-time evolution equation

$$x_{k+1} = f(x_k, u_k, w_k), \quad k = 0, 1, 2, \cdots, \tag{1}$$

where w_k is the external random disturbance at stage k. The random disturbances $w_k, k = 0, 1, 2, \cdots$, have identical statistics and are characterized by probabilities $P_W(\cdot|x_k, u_k)$ defined on the disturbance space D, where $P_W(w_k|x_k, u_k)$ is the probability of occupance of w_k, under the current state x_k and control u_k. The probability of w_k may depend explicitly on x_k and u_k but not on values of prior disturbances $w_{k-1}, \cdots, w_0$.

We consider the class of policies (also called control laws) that consist of a sequence of functions

$$\pi = \{\mu_0, \mu_1, \cdots, \},$$

where $\mu_k : S \to U, k = 0, 1, \cdots$ maps states x_k into controls $u_k = \mu_k(x_k)$ and in such that $\mu_k(x_k) \in U$ for all $x_k \in S$. Such policies will be called admissible. If a admissible policy $\pi = \{\mu_0, \mu_1, \cdots\}$ is given, the stochastic dynamical system (1) become a closed-loop dynamical system as follows

$$x_{k+1} = f(x_k, \mu_k(x_k), w_k), \tag{2}$$

where the control input $\mu_k(x_k)$ at time k is implemented by feedback of current state x_k.

Given an initial state x_0, and an admissible policy $\pi = \{\mu_0, \mu_1, \cdots\}$ consider the total expected cost

$$J_\pi(x_0) = \lim_{N\to\infty} \underset{\substack{w_k \\ k=0,1,\cdots}}{E} \sum_{k=0}^{N-1} \alpha^k g(x_k, u_k), \tag{3}$$

subject to the system equation constraint (2). Where $g : S \times U \to R$ is the per-step cost function and $0 < \alpha < 1$ is the discount factor.

We denote by Π the set of all admissible policies π, that is, the set of all sequences of functions $\pi = \{\mu_0, \mu_1, \cdots\}$. The purpose of the infinite horizon optimal control is to minimize $J_\pi(x_0)$ for a given x_0 over all $\pi \in \Pi$. The optimal cost function J^* is defined by

$$J^*(x) = \inf_{\pi\in\Pi} EJ_\pi(x), \quad x \in S. \tag{4}$$

Now, we give a simple example of stochastic Boolean logical dynamical system with two random disturbances.

Example 2.1. We consider in classical Boolean logical domain, that is the logical state space S and control space U both consist of two elements $S = U = \{T = 1, F = 0\}$. Assume that the disturbance space also consists of two elements $W = \{w^1 = T, w^2 = F\}$. A stochastic logical dynamical system is described by the following stationary evolution equation

$$x_{k+1} = f(x_k, u_k, w_k),$$

in which the logical function $f : S \times U \times W \to S$ is defined by

$$f(x, u, w) = (x \wedge u \wedge w) \vee [(x \to u) \wedge \neg w]\,. \tag{5}$$

Here the standard logical operations $\neg, \wedge, \vee, \to$ are negation, conjunction, disjunction and implication, respectively. The truth table of these operations are as follows Rade [1998]:

Table 1. Truth table for $\neg x, x \wedge u, x \vee u, x \to u$

x	u	$\neg x$	$x \wedge u$	$x \vee u$	$x \to u$
T	T	F	T	T	T
T	F	F	F	T	F
F	T	T	F	T	T
F	F	T	F	F	T

The conditional probabilities of occupance of disturbance w given current state and control is given below

$P_W(w^1|x = T, u = T) = 0.9, P_W(w^2|x = T, u = T) = 0.1,$
$P_W(w^1|x = T, u = F) = 0.2, P_W(w^2|x = T, u = F) = 0.8,$
$P_W(w^1|x = F, u = T) = 0.4, P_W(w^2|x = F, u = T) = 0.6,$
$P_W(w^1|x = F, u = F) = 0.3, P_W(w^2|x = F, u = F) = 0.7.$

Define the per-step cost function $g : S \times U \to R$ as follows

$$g(x = T, u = T) = 2, g(x = T, u = F) = 3, \tag{6}$$
$$g(x = F, u = T) = 1, g(x = F, u = F) = 1.$$

Remark 2.1. Since the disturbance space just include two element in above example, the logical function (5) and the conditional probabilities of disturbance can be enumerated in a truth table (see Table 2). But in the general case, it is impossible to uniformly represent the value of logical function and the conditional probabilities of disturbance by enumeration.

Table 2. Truth table for logical function (5) and conditional probabilities P_w for disturbance w.

x	u	w	f	P_w
T	T	T	T	0.9
F	T	T	F	0.4
T	F	T	F	0.2
F	F	T	T	0.3
T	T	F	T	0.1
F	T	F	F	0.6
T	F	F	F	0.8
F	F	F	T	0.7

3. MATRIX EXPRESSION OF MODEL

For convenience of description, we first give some notations:

(i) The set of $m \times n$ real matrices is denoted by $\mathcal{M}_{m\times n}$.

(ii) Let M be a matrix. Then $Col_i(M)(Row_i(M))$ denotes the $i-$th column (row) of M.

(iii) δ_s^i denote the i-column of the identity matrix I_s. And set

$$\Delta_s := \{\delta_s^i | i = 1, 2, \cdots, s\}.$$

(iv) A matrix $L \in \mathcal{M}_{m\times n}$ is called a logical matrix if its columns $Col(L) \subset \Delta_m$. Then any logical matrix L has the form $L = [\delta_m^{i_1}, \delta_m^{i_2}, \cdots, \delta_m^{i_n}]$, and briefly denoted as

$$L = \delta_m[i_1, i_2, \cdots, i_n].$$

The set of $m \times n$ logical matrices is denoted by $\mathcal{L}_{m\times n}$.

First, we identify the state space S with Δ_s as follows:

$$x^i \sim \delta_s^i, \quad i = 1, 2, \cdots, s. \tag{7}$$

Then, each element $x \in S$ has its corresponding vector form (still use x) $x \in \Delta_s$. Similarly, we identify the control space U with Δ_r as follows:

$$u^j \sim \delta_r^j, \quad j = 1, 2, \cdots, r. \tag{8}$$

3.1 Semi-tensor product of matrix

Semi-tensor product of matrices is a generalization of conventional matrix product [Cheng et al., 2011]. It has been successfully used for both continuous time dynamic systems [Mei et al., 2010] and discrete time logical dynamic system [Cheng et al., 2012]. We refer to the following books Cheng et al. [2012] for details.

Definition 3.1. Let $M \in \mathcal{M}_{m\times n}, N \in \mathcal{M}_{p\times q}$. The semi-tensor product of M and N, denoted as $M \ltimes N$, is defined by

$$M \ltimes N := (M \otimes I_{s/n})(N \otimes I_{s/p}), \tag{9}$$

where $s = lcm\{n, p\}$ is the least common multiple of n and p; $\otimes$ is the kronecker product.

Remark 3.1. (i) When $n = p$, the semi-tensor product and the conventional matrix product coincide. Hence the semi-tensor product is a generalization of conventional matrix product to any two matrices.

(ii) All the properties of conventional matrix product remain true for this generalization.

Here, we give some special cases to illustrate the definition of semi-tensor product.

Example 3.1. (1) Let $x = [x_1, x_2, \cdots, x_m] \in R^m, y = [y_1, y_2, \cdots, y_n] \in R^n$, Then, the semi-tensor product $x \ltimes y$ is

$$[x_1y_1, x_2y_1, \cdots, x_my_1, \cdots, x_1y_n, \cdots, x_my_n] \in R^{mn}.$$

(2) Let X be a row vector of dimension np , and Y be a column vector of dimension p . Then we split into p equal-size blocks as $X^1, \cdots, X^p$, which are $1 \times n$ rows. Then, the semi-tensor product $x \ltimes y$ is

$$x \ltimes y = \sum_{i=1}^{p} X^i y_i \in R^n.$$

3.2 Markov process description of stochastic dynamics

Since the logical state space and control space are both consist of finite logical elements, the stochastic logical dynamical system with control often conveniently specified in term of a finite-state Markov process with control. We denote by $p_{ij}(\delta_r^l)$ the transition probabilities from state δ_i using control $u = \delta_r^l$ to the next state δ_j,

$$p_{ij}(\delta_r^l) = P(x_{k+1} = \delta_s^j | x_k = \delta_s^i, u_k = \delta_r^l), \tag{10}$$

for all $\delta_s^i, \delta_s^j \in S, \delta_r^l \in U$. It was noticed that the transition probabilities $p_{ij}(\delta_r^l)$ satisfy

$$\sum_{j=1}^{s} p_{ij}(\delta_r^l) = 1, \quad \forall\ i = 1, \cdots, s, \quad l = 1, \cdots, r.$$

Then, given a discrete-time evolution system in the form (1) together with the probability distribution $P_W\{w|x_k, u_k\}$ of w_k, we can provide an equivalent Markov process description. The corresponding transition probability is given by

$$p_{ij}(\delta_r^l) = P_W(W_{ij}(u)|\delta_s^i, \delta_r^l) \tag{11}$$

where $W_{ij}(\delta_r^l)$ is the set

$$W_{ij}(\delta_r^l) = \{w : f(\delta_s^i, \delta_r^l, w) = \delta_s^j\}.$$

For the fixed control $\delta_r^k \in U$, we denote by $P^k \in \mathcal{M}_{s\times s}$ the transition probability matrix

$$P^k = \begin{pmatrix} p_{11}(\delta_r^k) & \cdots & p_{1s}(\delta_r^k) \\ p_{21}(\delta_r^k) & \cdots & p_{2s}(\delta_r^k) \\ \vdots & \vdots & \vdots \\ p_{s1}(\delta_r^k) & \cdots & p_{ss}(\delta_r^k) \end{pmatrix}, \quad k = 1, \cdots, r. \tag{12}$$

For statement ease, we collect the all transition probability matrices together, and define the matrix $\mathbb{P} \in \mathcal{M}_{(rs)\times s}$ by

$$\mathbb{P} = \begin{pmatrix} P^1 \\ P^2 \\ \vdots \\ P^r \end{pmatrix}. \tag{13}$$

Denoting

$$g(\delta_s^i, \delta_r^j) = g_{ij}, \quad i = 1, \cdots, s; j = 1, \cdots, r,$$

we define a matrix, called the per-step cost matrix, as

$$G = (g_{ij}) \in \mathcal{M}_{s\times r}.$$

So, we have

$$g(x, u) = x^T G u, \qquad \forall x \in \Delta_s, u \in \Delta_r, \tag{14}$$

and the objective function (4) becomes, for any $x_0 \in \Delta_s$

$$J^*(x_0) = \inf_{\pi\in\Pi} \lim_{N\to\infty} E \sum_{k=0}^{N-1} \alpha^k x_k^T G \mu_k(x_k), \tag{15}$$

Example 3.2. Now we give the matrix expression of Example 2.1. To use matrix expression we identify logical values as

$$T \equiv \delta_2^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad F \equiv \delta_2^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Then according to logical operation (5) and formula (11), we get that the transition probability matrices corresponding to the controls $u = \delta_2^1$ and $u = \delta_2^2$ are

$$P^1 = P(\delta_2^1) = \begin{pmatrix} p_{11}(\delta_2^1) & p_{12}(\delta_2^1) \\ p_{21}(\delta_2^1) & p_{22}(\delta_2^1) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0.6 & 0.4 \end{pmatrix},$$

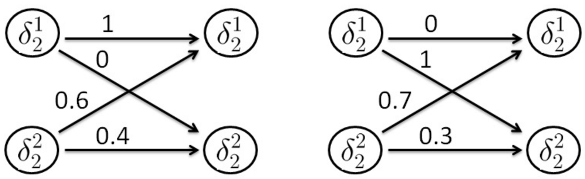
$$P^2 = P(\delta_2^2) = \begin{pmatrix} p_{11}(\delta_2^2) & p_{12}(\delta_2^2) \\ p_{21}(\delta_2^2) & p_{22}(\delta_2^2) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0.7 & 0.3 \end{pmatrix}.$$

The corresponding state transition diagram is shown in Fig. 1. Moreover, we have

$$\mathbb{P} = \begin{pmatrix} P^1 \\ P^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0.6 & 0.4 \\ 0 & 1 \\ 0.7 & 0.3 \end{pmatrix}. \tag{16}$$

The matrix expression of the one-step cost function $g : S \times U \to R$ is $g(x, u) = x^T G u$ with

$$G = \begin{pmatrix} 2 & 3 \\ 1 & 2 \end{pmatrix}. \tag{17}$$



(a) When $u = \delta_2^1$. (b) When $u = \delta_2^2$.

Fig. 1. State transition diagram for Example 3.2.

4. MATRIX EXPRESSION OF DYNAMIC PROGRAMMING

We denote by $\mathcal{U}$ the set of all control law $\mu : S \to U$, that is

$$\mathcal{U} = \{\mu | \mu : S \to U\}.$$

Since the state space S and control space U are both finite, it is obviously that the capacity of $\mathcal{U}$ is $|\mathcal{U}| = r^s$.

Under the logical vector setting, we can see the control law as a logical operator form Δ_s to Δ_r. So the following result is fundamental.

Proposition 4.1. For any $\mu \in \mathcal{U}$, there exists a unique logical matrix $\Phi_\mu \in \mathcal{L}_{r\times s}$, called the structure matrix of μ, such that in the vector form μ is expressed as

$$\mu(x) = \Phi_\mu(x), \qquad \forall\ x \in \Delta_s. \tag{18}$$

The optimal cost functional for the one-stage problem that has stage cost g and terminal cost of J, denoted by $TJ(x)$ with

$$TJ(x) = \inf_u \mathop{E}_w \{g(x, u) + J(f(x, u, w))\}, \quad x \in \Delta_s. \tag{19}$$

Similarly, for any function J on Δ_s and any control function $\mu : \Delta_s \to \Delta_r$, we denote

$$T_\mu J(x) = \underset{w}{E}\{g(x,u) + J(f(x,\mu(x),w))\}, \quad x \in \Delta_s. \quad (20)$$

$T_\mu J$ may be viewed as the cost function associated with control law $\mu : \Delta_s \to \Delta_r$ for the one stage problem that has stage cost g and terminal cost J. For all $k = 0, 1, 2, \cdots$, we write

$$(T^0 J)(x) = J(x), \;\; x \in \Delta_s,$$
$$(T^k J)(x) = (T(T^{k-1}J))(x), \;\; x \in \Delta_s.$$

Similarly, $T_\mu^k J$ is defined by

$$(T^0 J_\mu)(x) = J(x), \;\; x \in \Delta_s,$$
$$(T_\mu^k J)(x) = (T_\mu(T_\mu^{k-1}J))(x), \;\; x \in \Delta_s.$$

Any function J on Δ_s, as well as the functions TJ and $T_\mu J$ may be represented by the $s-$dimensional vectors

$$J = \begin{pmatrix} J(\delta_s^1) \\ \vdots \\ J(\delta_s^s) \end{pmatrix}, T_\mu J = \begin{pmatrix} T_\mu J(\delta_s^1) \\ \vdots \\ T_\mu J(\delta_s^s) \end{pmatrix} \quad (21)$$

$$TJ = \begin{pmatrix} TJ(\delta_s^1) \\ \vdots \\ TJ(\delta_s^s) \end{pmatrix} = \begin{pmatrix} \inf_\mu \{T_\mu J(\delta_s^1)\} \\ \vdots \\ \inf_\mu \{T_\mu J(\delta_s^s)\} \end{pmatrix} \quad (22)$$

For a control law μ, we denote by P_μ the transition probability matrix

$$P_\mu = \begin{pmatrix} p_{11}(\mu(\delta_1)) & \cdots & p_{1s}(\mu(\delta_1)) \\ \vdots & \vdots & \vdots \\ p_{s1}(\mu(\delta_n)) & \cdots & p_{ss}(\mu(\delta_n)) \end{pmatrix}, \quad (23)$$

by g_μ the cost vector

$$g_\mu = \left(g(\delta_s^1, \mu(\delta_s^1)), \cdots, g(\delta_s^s, \mu(\delta_s^s))\right)^T.$$

Proposition 4.2. For any control law $\mu : S \to U$, define the matrix $M_\mu \in \mathcal{M}_{s\times(sr)}$ as

$$M_\mu = \begin{pmatrix} (\delta_s^1)^T \ltimes (\mu(\delta_s^1))^T \\ (\delta_s^2)^T \ltimes (\mu(\delta_s^2))^T \\ \vdots \\ (\delta_s^s)^T \ltimes (\mu(\delta_s^s))^T \end{pmatrix}. \quad (24)$$

Then the transition probability matrix P_μ associated with stationary policy μ can be calculated by

$$P_\mu = M_\mu \mathbb{P}. \quad (25)$$

Proof. It is enough to prove that, for any $i = 1, \cdots, s$

$$Row_i(M_\mu \mathbb{P}) = Row_i(P_\mu). \quad (26)$$

We assume that $\mu(\delta_s^i) = \delta_r^j$. Then,

$$\begin{aligned} Row_i(M_\mu \mathbb{P}) &= (\delta_s^i)^T \ltimes (\mu(\delta_s^i))^T \mathbb{P} \\ &= (\delta_s^i)^T \ltimes (\delta_r^j)^T \mathbb{P} \\ &= (\delta_s^i)^T \left((\delta_r^j)^T \otimes I_s\right) \mathbb{P} \\ &= \delta_{sr}^{(j-1)s+i} \mathbb{P} = Row_{(j-1)s+i}(\mathbb{P}) \\ &= Row_i\left(P^j\right). \end{aligned}$$

On the other hand, if $\mu(\delta_s^i) = \delta_r^j$, by definition of P_μ, we get, for all $i = 1, \cdots, s$

$$Row_i(P_\mu) = Row_i\left(P^j\right).$$

Thus we prove that (26).

Definition 4.1. We define a hyperplane $\mathbb{D}^{s+1}$ of R^{s+1} as

$$\mathbb{D}^{s+1} = \{(x_0, x_1, \cdots, x_s) \in R^{s+1} : x_0 = 1\}.$$

We define an increasing dimension operator $\hat{\cdot} : R^s \to R^{s+1}$, as $\hat{J} = (1, J^T)^T$, for all $J \in R^s$. And, for any function J on Δ_s, we define

$$Q\hat{J} = \inf_\mu Q_\mu \hat{J}, \text{ with } Q_\mu = \begin{pmatrix} 1 & 0 \\ g_\mu & P_\mu \end{pmatrix}. \quad (27)$$

According to the definition of Q_μ, we immediately obtain

Proposition 4.3. For any function J on Δ_s, we have

$$\widehat{T_\mu J} = Q_\mu \hat{J} \text{ and } \widehat{TJ} = Q\hat{J} \quad (28)$$

Proof. For given policy $\mu : S \to U$, according to the definition of operator Q_μ, we get

$$\begin{aligned} Q_\mu \hat{J} &= \begin{pmatrix} 1 & 0 \\ g_\mu & P_\mu \end{pmatrix} \begin{pmatrix} 1 \\ J \end{pmatrix} = \begin{pmatrix} 1 \\ g_\mu + P_\mu J \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ T_\mu J(\delta_s^1) \\ \vdots \\ T_\mu J(\delta_s^s) \end{pmatrix} = \begin{pmatrix} 1 \\ T_\mu J \end{pmatrix} = \widehat{T_\mu J}. \end{aligned}$$

Similarly, one can prove that $\widehat{TJ} = Q\hat{J}$.

Now we give the algebraic expression of $Q\hat{J}$.

Theorem 4.1. For any function J on Δ_s, $Q\hat{J}$ can be calculated by following formula

$$Q\hat{J} = \begin{pmatrix} 1 \\ \min_{j=1,\cdots,r} \{(\delta_s^1)^T G \delta_r^j + (\delta_s^1)^T \ltimes (\delta_r^j)^T \mathbb{P} J\} \\ \min_{j=1,\cdots,r} \{(\delta_s^2)^T G \delta_r^j + (\delta_s^2)^T \ltimes (\delta_r^j)^T \mathbb{P} J\} \\ \vdots \\ \min_{j=1,\cdots,r} \{(\delta_s^s)^T G \delta_r^j + (\delta_s^s)^T \ltimes (\delta_r^j)^T \mathbb{P} J\} \end{pmatrix}, \quad (29)$$

and the one-step optimal control policy $\mu^* : S \to U$ for $Q\hat{J}$ is

$$\mu^*(x) = \underset{j\in\{1,\cdots,r\}}{\arg\min} \{xG\delta_r^j + x^T \ltimes (\delta_r^j)^T \mathbb{P} J\}, \quad (30)$$

for any $x \in S$.

Proof. According to definition of operator Q and equation (20), we get

$$\begin{aligned} Q\hat{J} = \widehat{TJ} &= \begin{pmatrix} 1 \\ TJ \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ \inf_{u\in U} \underset{w}{E}\{g(\delta_s^1, u) + J(f(\delta_s^1, u, w))\} \\ \inf_{u\in U} \underset{w}{E}\{g(\delta_s^2, u) + J(f(\delta_s^2, u, w))\} \\ \vdots \\ \inf_{u\in U} \underset{w}{E}\{g(\delta_s^s, u) + J(f(\delta_s^s, u, w))\} \end{pmatrix}. \end{aligned}$$

We also notice that, for any $i = 1, 2, \cdots, s$,

$$\begin{aligned}
&\inf_{u\in U} \mathop{E}_{w} \left\{g(\delta_s^s,u)+J(f(\delta_s^i,u,w))\right\}\\
&=\inf_{\delta_r^j\in U} \mathop{E}_{w} \left\{g(\delta_s^s,\delta_r^j)+J(f(\delta_s^s,\delta_r^j,w))\right\}\\
&=\inf_{j=1,\cdots,r}\left\{g(\delta_s^s,\delta_r^j)+\sum_{k=1}^{s}p_{ik}(\delta_r^j)J(\delta_s^k)\right\}\\
&=\min_{j=1,\cdots,r}\{(\delta_s^1)^TG\delta_r^j+(\delta_s^1)^T\ltimes(\delta_r^j)^T\mathbb{P}J\},
\end{aligned}$$

by the definition of matrix $\mathbb{P}$ and semi-tensor product. So, we finish the proof of Theorem 4.2.

Lemma 4.1. For any control law $\mu:S\to U$, $\lambda=1$ is a simple eigenvalue of Q_μ.

Proof. See the Appendix.

Proposition 4.4. For any control law $\mu:S\to U$, $Q_\mu:\mathbb{D}^{s+1}\to\mathbb{D}^{s+1}$ admits a unique fixed point, that is there exists a unique point $J_\mu\in R^n$ such that

$$Q_\mu\begin{pmatrix}1\\J_\mu\end{pmatrix}=\begin{pmatrix}1\\J_\mu\end{pmatrix}. \tag{31}$$

Proof. See the Appendix.

Proposition 4.5. For any control law $\mu:\Delta_s\to\Delta_r$, for any vector $J\in R^s$, the operator $Q_\mu:\mathbb{D}^{s+1}\to\mathbb{D}^{s+1}$ and the corresponding cost vector $J_\mu\in R^s$ of μ satisfy

$$\lim_{N\to\infty}Q_\mu^N\hat{J}=\hat{J}_\mu. \tag{32}$$

Proof. We notice that

$$Q_\mu(\hat{J}-\hat{J}_\mu)=\begin{pmatrix}1&0\\g_\mu&\alpha P_\mu\end{pmatrix}\begin{bmatrix}0\\\hat{J}-\hat{J}_\mu\end{bmatrix}=\begin{bmatrix}0\\\alpha P_\mu(\hat{J}-\hat{J}_\mu)\end{bmatrix},$$

Hence, by iteration, for any N,

$$\begin{aligned}
Q_\mu^N(\hat{J}-\hat{J}_\mu)&=Q_\mu^{N-1}Q_\mu(\hat{J}-\hat{J}_\mu)=Q_\mu^{N-1}\begin{bmatrix}0\\\alpha P_\mu(\hat{J}-\hat{J}_\mu)\end{bmatrix}\\
&=\begin{bmatrix}0\\\alpha^NP_\mu^N(\hat{J}-\hat{J}_\mu)\end{bmatrix}.
\end{aligned}$$

Since $0<\alpha<1$, we get

$$\lim_{N\to\infty}\|Q_\mu^N(\hat{J}-\hat{J}_\mu)\|\le\lim_{N\to\infty}\alpha^N\|P_\mu\|\|J-J_\mu\|=0.$$

So, according to Proposition 4.4,

$$\lim_{N\to\infty}Q_\mu^N\hat{J}=\lim_{N\to\infty}Q_\mu^N\hat{J}_\mu=\hat{J}_\mu. \tag{33}$$

Finally, we give the matrix expression of dynamic programming for the stochastic logical optimal control problem.

Theorem 4.2. (Value iteration approximation) For any function $J:\Delta_s\to R$, the optimal cost vector $J^*=\left(J^*(\delta_s^1),J^*(\delta_s^2),\cdots,J^*(\delta_s^2)\right)^T$ satisfies

$$\hat{J}^*=\lim_{N\to\infty}Q^N\hat{J}, \tag{34}$$

and

$$\|\hat{J}^*-Q^N\hat{J}\|\le\alpha^k\left(\frac{M_1}{1-\alpha}+M_2\right), \tag{35}$$

where $Q^N\hat{J}=Q(Q^{N-1}\hat{J})$, $M_1=\max_{i,j}g_{ij}$ and $M_2=\max_{\delta_s^i\in\Delta_s}J(\delta_s^i)$.

Example 4.1. Continue Example 3.2. If given a terminal cost function $J:\Delta_2\to R$ as

$$J(\delta_2^i)=[3\ 1]\delta_2^i, i=1,2, \tag{36}$$

then, according to Theorem 4.2, we can calculate

$$\begin{aligned}
&\inf_{u\in\Delta_2}\{\delta_2^1Gu+(\delta_2^1)^T\ltimes u^T\mathbb{P}J\}\\
&=\min\left\{\begin{array}{l}\delta_2^1G\delta_2^1+(\delta_2^1)^T\ltimes(\delta_2^1)^T\mathbb{P}J,\\\delta_2^1G\delta_2^2+(\delta_2^1)^T\ltimes(\delta_2^2)^T\mathbb{P}J\end{array}\right\}\\
&=\min\{g_{11}+(1,0,0,0)\mathbb{P}J,g_{12}+(0,0,1,0)\mathbb{P}J\}\\
&=\min\{3.8,5.7\}=3.8,
\end{aligned}$$

that means

$$\mu_0^*(\delta_2^1)=\delta_2^1,\quad J_0^*(\delta_2^1)=3.8.$$

Similarly we also get

$$\mu_0^*(\delta_2^2)=\delta_2^1,\quad J_0^*(\delta_2^1)=3.16,$$

Hence

$$Q\hat{J}=\begin{pmatrix}1\\3.8\\3.16\end{pmatrix}. \tag{37}$$

Continuing this process, we obtain that

$$\begin{aligned}
\mu_{99}^*(\delta_2^1)&=\delta_2^2,\quad J_{99}^*(\delta_2^1)=18.3112,\\
\mu_{99}^*(\delta_2^2)&=\delta_2^1,\quad J_{99}^*(\delta_2^1)=7.0125.\\
\mu_{100}^*(\delta_2^1)&=\delta_2^2,\quad J_{100}^*(\delta_2^1)=18.31127,\\
\mu_{100}^*(\delta_2^2)&=\delta_2^1,\quad J_{100}^*(\delta_2^1)=17.0126.
\end{aligned}$$

Fig. 2 shows the dynamical programming approximation result with stage N from 1 to 100.

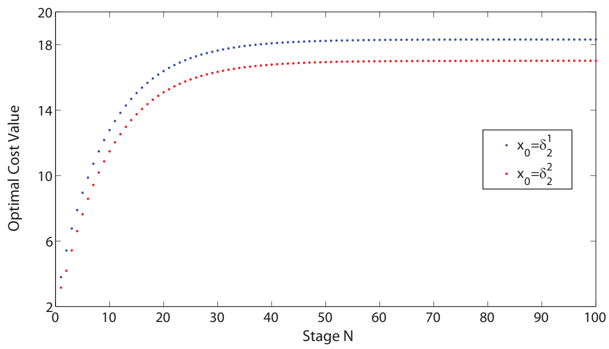


Fig. 2. Dynamical programming approximation result of Example 4.1.

Theorem 4.3. I. (Bellman's Equation) The optimal cost function J^* satisfies

$$\hat{J}^*=Q\hat{J}^* \tag{38}$$

II. (Necessary and Sufficient Condition for Optimality) A stationary policy μ is optimal if and only if

$$Q\hat{J}_\mu=Q_\mu\hat{J}_\mu \tag{39}$$

Example 4.2. Continue Example 3.2 and Example 4.1. Since $s=r=2$, there are 4 control laws μ_1,μ_2,μ_3,μ_4 in $\mathcal{U}$, and the corresponding structure matrix of these control laws are denoted as

$$\Phi_{\mu_1}=\delta_2[1,1],\ \Phi_{\mu_2}=\delta_2[1,2],\ \Phi_{\mu_3}=\delta_2[2,1],\ \Phi_{\mu_4}=\delta_2[2,2],$$

respectively. Then, based on Proposition 4.4, we get

$$J_{\mu_1} = \begin{bmatrix} 20 \\ 18.4375 \end{bmatrix}, J_{\mu_2} = \begin{bmatrix} 20 \\ 20 \end{bmatrix},$$
$$J_{\mu_3} = \begin{bmatrix} 18.3117 \\ 17.013 \end{bmatrix}, J_{\mu_4} = \begin{bmatrix} 24.4785 \\ 23.865 \end{bmatrix},$$

It is clear that the optimal control law is μ_3, that is

$$\mu_3(\delta_2^1) = \delta_2[2,1]\delta_2^1 = \delta_2^2 \text{ and } \mu_3(\delta_2^2) = \delta_2[2,1]\delta_2^2 = \delta_2^1,$$

and the corresponding optimal cost functions are

$$J^*(\delta_2^1) = 18.3117, \quad J^*(\delta_2^2) = 17.013,$$

which is coincide with the dynamical programming approximation result given by Example 4.1.

5. CONCLUSION

This paper considered the infinite horizon optimal control problem of stochastic logical control dynamical system with finite state. The infinite horizon optimization problem was formulated in algebraic form, after giving two equivalent descriptions of the stochastic logical control dynamical system. Based on semi-tensor product of matrices and increasing dimensional technique, we established the matrix expression of dynamic programming for the optimal control problem. Some examples were presented to illustrate the theoretical results.

6. APPDEDIX

Proof of Lemma 4.1 Obvious $\lambda = 1$ is an eigenvalue of Q_μ, since

$$det\,(I_{s+1} - Q_\mu) = det \begin{pmatrix} 0 & 0 \\ -g_\mu & I_s - \alpha P_\mu \end{pmatrix} = 0. \tag{40}$$

Moreover, we can prove that

$$dim\,(R(I_{s+1} - Q_\mu)) = n. \tag{41}$$

Since $\alpha < 1$ and $\sum_{j=1}^{s} P_\mu(\mu) = 1$, for all $j = 1, \cdots, s$, we get the matrix

$$I_s - \alpha P_\mu \tag{42}$$

is inverse. So for any $y \in R^s$, we take

$$x = (I_s - \alpha P_\mu)^{-1}(y + g_\mu), \tag{43}$$

then

$$(I_{s+1} - Q_\mu) \begin{pmatrix} 0 \\ x \end{pmatrix} = \begin{pmatrix} 0 \\ y \end{pmatrix}, \tag{44}$$

that means, for all $y \in R^s$,

$$\begin{pmatrix} 0 \\ y \end{pmatrix} \in R(I_{s+1} - Q_\mu) \tag{45}$$

Hence, $dim\,(R(I_{s+1} - Q_\mu)) = n$. Finally, according to the rank-nullity (dimension) theorem, we get

$$\begin{aligned} dim(Ker(I_{s+1} - Q_\mu)) &= (s+1) - dim\,(R(I_{s+1} - Q_\mu)) \\ &= 1. \end{aligned}$$

Therefore, $\lambda = 1$ is a simple eigenvalue of Q_μ and the proof of Lemma 4.1 is completed.

Proof of Proposition 4.4 Let ξ_μ is the eigenvector of matrix Q_μ corresponding to the eigenvalue $\lambda = 1$ with $\|\xi_\mu\| = 1$. Since $\lambda = 1$ is a simple eigenvalue of Q_μ, by Lemma 4.1, we know that ξ_μ is unique. Set

$$\xi_\mu = \begin{pmatrix} \xi_\mu^0 \\ \eta_\mu \end{pmatrix} \quad \text{with} \quad \eta_\mu = \left(\xi_\mu^1, \cdots, \xi_\mu^s\right)^T,$$

where ξ_μ^{n-1} is the n-th component of the eigenvector ξ_μ, respectively, $n = 1, 2, \cdots, s+1$.

We claim that $\xi_\mu^0 \neq 0$. By contradiction, suppose $\xi_\mu^0 = 0$. Then, by

$$\begin{aligned} 0 = (I_{s+1} - Q\mu)\xi_\mu &= \begin{pmatrix} 0 & 0 \\ -g(\mu) & I_s - \alpha F_\mu \end{pmatrix} \begin{pmatrix} 0 \\ \eta_\mu \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ (I_s - \alpha F_\mu)\eta_\mu \end{pmatrix}, \end{aligned}$$

and the matrix $I_s - \alpha F_\mu$ is inverse, we obtain $\eta_\mu = 0$. That means $\xi_\mu = ((\xi_\mu^0)^T, (\eta_\mu)^T)^T = 0$, this is contrary to $\|\xi_\mu\| = 1$, So $\xi_\mu^0 \neq 0$. Finally, let $J_\mu = \frac{1}{\xi_\mu^0}$, then J_μ is unique and satisfies the equation (31). We finish the proof of Proposition 4.4.

REFERENCES

T. Akutsu, M. Hayashida, W. Ching, M.K. Ng. Control of Boolean networks: hardness results and algorithms for tree structured networks, *J. Theoretical Biology,* 670-679, 244, 2007.

R. Bellman. *Dynamic Programming*, Princeton Univ. Press, NJ, 1957.

D. P. Bertsekas. *Dynamic programming and optimal control* (Vol. 1, Vol.2). Belmont: Athena Scientific, 1995.

D. Cheng, H. Qi, Z. Li. *Analysis and Control of Boolean Networks { A Semi-tensor Product Approach}*, Springer, London, 2011.

D. Cheng, H. Qi, Y. Zhao. *An Introduction to Simetensor Product of Matrices and Its Applications*, World Scientific, Singapore, 2012.

E. Doberkat. A Stochastic Interpretation of Propositional Dynamic Logic: Expressivity, *Logic and Its Applications,* Lecture Notes in Computer Science, 50-64, 6521, 2011.

S. Karaman, R.G. Sanfelice, E. Frazzoli, Optimal Control of Mixed Logical Dynamical Systems with Linear Temporal Logic Specifications, *Conference on Decision and Control,* 2117-2122, 2008.

B. Faryabi, A. Datta, E. R. Dougherty. On approximate stochastic control in genetic regulatory networks, *IET Systems Biology*, 361-368, 1 (6), 2007.

F. Li, J. Sun., Stability and stabilization of multivalued logical networks, *Nonlinear Analysis: Real World Applications,* 3701-3712, 12 (6), 2011.

S. Mei, F. Liu, A. Xue. *Semi-tensor Product Approach to the Transient Analysis of Power Systems*, Tsinghua Univ. Press, Beijing, 2010 (in Chinese).

R. Oal, A. Datta, E.R. Dougherty, Optimal infinite-horizon control for probabilistic Boolean networks, *IEEE Trans. Signal Proc.,* 54 (6), 2375-2387, 2006.

L. Rade, B. Westergren. *Mathematics Handbook for Science and Engineering,* 4th ed, Studentlitteratur, Sweden, 1998.

K. Truemper. *Design of Logic-based Intelligent Systems,* Wiley & Sons, New Jersey, 2004.

Y. Zhao, Z. Li, D. Cheng, Optimal control of logical control networks, *IEEE Trans. Aut. Contr.,* 1766-1776, 56 (8), 2011.

Y. Wu, T. Shen. A Logical Dynamical Systems Approach to Modeling and Control of Residual Gas Fraction in IC Engines, *7th IFAC Symposium on Advances in Automotive Control,* pp. 495-500, 2013.