2B2-6

Analysis and Design of Discrete-Event Dynamic Systems Based on Recurrence Inequalities

*Yoshifumi Okuyama
Humanitech Laboratory & Tottori University (Emeritus)

Abstract There are many event-driven types of discrete control systems e.g., manufacturing systems, industry and welfare robots, and even (packet-based) communication networks. However, those dynamic systems is highly complex and hybrid-type structures. Therefore, analysis and design of such discrete control systems should be treated as logical and really 'discrete' dynamic systems. In this paper, the preferable and robust performance of DEDSs is studied by using state transision vectors and permutation matrices. And the optimality (preferability) of system performance is discussed based on the recurrent inequalities expression of dynamic programming.

Key Words: Discrete event systems, Permutation matrices, Multiple metrics, Nonnegative inverse matrices, Dynamic programming

1 Introduction

Nowadays, there are many event-driven types of discrete control systems (i.e., discrete-event dynamic systems, DEDSs [1, 2, 3, 4]),e.g., manufacturing systems, industry and welfare robots, and even (packet-based) communication networks for IoT [5, 6]. However, those dynamic systems is highly complex and hybrid-type structures. Therefore, analysis and design of such discrete control systems should be treated as logical and really 'discrete' dynamic systems[1)]. In this paper, the preferable and robust performance of DEDSs will be investigated by using state transferred vectors and permutation matrices. The optimality (preferability) of system performance will be discussed based on the recurrent inequalities expression of *Dynamic Programming* [7, 8].

2 Mathematical Description of DEDSs

In general, finite-state and discrete-event dynamic systems (DEDSs) can be written as the following multistage processes

$$\boldsymbol{x}(t_{k+1}) = \boldsymbol{f}(\boldsymbol{x}(t_k), \boldsymbol{e}(t_k)) \qquad (1)$$
$$k \in \mathbb{N} := \{0, 1, 2, \ldots 1, N\},$$

where $\boldsymbol{x}(\cdot)$, $\boldsymbol{e}(\cdot)$, and $\boldsymbol{f}(\cdot,\cdot)$ are states, event-signals, and a transition function, respectively. In this paper, each variable is considered as

$$\boldsymbol{x}(t_k) \in \mathbb{Z}^n, \quad \boldsymbol{e}(t_k) \in \mathbb{Z}^m, \boldsymbol{f}: \ \mathbb{Z}^n \times \mathbb{Z}^m \to \mathbb{Z}^n,$$

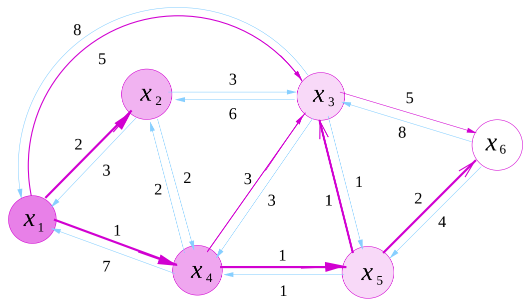

Fig. 1: Packet-switching communication network.

where $\mathbb{Z}$ is a finite set of integers. Of course, the states and events may be non-numerical (qualitative) situations in practice. However, they can be considered as ordered sets, and thus the ordered sets will be replaced by integer numbers.

With respect to simple DEDSs, (1) can also be written by using a permutation matrix [9, 10, 11] (connective (0,1)-matrix) as

$$x(t_{k+1}) = \boldsymbol{\mathcal{P}}(\boldsymbol{x}(t_k), \boldsymbol{e}(t_k)) \cdot x(t_k), \qquad (2)$$

where [2)]

$$\boldsymbol{\mathcal{P}}(\cdot,\cdot) = \boldsymbol{P}(t_k) \in \mathbb{P} \subset \mathbb{I}_+^{n\times n}$$
$$\mathbb{P}: \text{permutation } (0,1)\text{-matrix}, \quad \mathbb{I}_+ := \{0, 1\}.$$

Therefore, (2) can be expanded as follows[3)] :

$$\boldsymbol{x}(t_k) = \left(\prod_{h=0}^{k-1} \boldsymbol{P}(t_h)\right) \boldsymbol{x}(t_0). \qquad (3)$$

1) At presesnt, every (computer) control system should be considered as a discrete dynamic system regardles of whether the process is time or event driven.

2) In the following, we write simply $\boldsymbol{\mathcal{P}}(t_k) = \boldsymbol{\mathcal{P}}(\boldsymbol{x}(t_k), \boldsymbol{e}(t_k))$.

3) In regard to the above (0,1)-matrix $\boldsymbol{\mathcal{P}} \in \mathbb{P}$, only one of '1' should be permitted in the raw elements. The number of '1' in the matrix $\mathbb{P}$ is n. That is, the addition of each state will be prohibitted.

Figure 1 shows an example of packet switching (communication) network. In the text-book [12], a shortest path from x_1 to x_6 problem was solved using Bellman-Ford algorithm (i.e., *Dynamic Programming* [13, 14])[4)].

In this case, the permutation matrices of the optimum solution (preferable route) are given below:

$$\boldsymbol{P}_{14} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{e_{14}},$$

$$\boldsymbol{P}_{45} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{e_{45}},$$

$$\boldsymbol{P}_{56} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{e_{56}}.$$

Therefore, the transferred state-vectors are expressed as follows:

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \end{bmatrix} \overset{e_{14}}{\rightarrow} \begin{bmatrix} \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{x}_1 \\ \hat{x}_5 \\ \hat{x}_6 \end{bmatrix} \overset{e_{45}}{\rightarrow} \begin{bmatrix} \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_1 \\ \hat{x}_6 \end{bmatrix} \overset{e_{56}}{\rightarrow} \begin{bmatrix} \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \\ \hat{x}_1 \end{bmatrix}. \tag{4}$$

As for the forward process from x_1 to x_6, the matrix transformation will be given as[5)]:

$$\boldsymbol{P}_f = \boldsymbol{P}_{56} \cdot \boldsymbol{P}_{45} \cdot \boldsymbol{P}_{14}. \tag{5}$$

On the other hand, as for the backward process from x_6 to x_1, the preferable route is given by the following matrices:

$$\boldsymbol{P}_{63} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{e_{63}},$$

$$\boldsymbol{P}_{32} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{e_{32}},$$

[4)] The derivation process will be described in section **3**.

[5)] Although the above matrices expressions for DEDs may seem to be ridiculous, the quantative performances of them cannot be clarified only using set and graph theory.

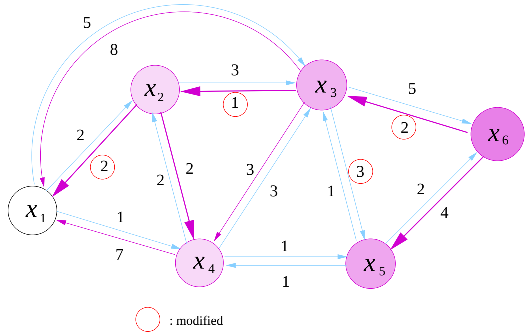

Fig. 2: Modified packet-switching network.

$$\boldsymbol{P}_{21} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{e_{21}}.$$

In this example, since the costs of backward route is defined, the above matrices can be obtained.

Therefore, the transferred state-vectors are written as follows:

$$\begin{bmatrix} \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \\ \hat{x}_1 \end{bmatrix} \overset{e_{63}}{\rightarrow} \begin{bmatrix} \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_1 \\ \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \end{bmatrix} \overset{e_{32}}{\rightarrow} \begin{bmatrix} \hat{x}_2 \\ \hat{x}_1 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \end{bmatrix} \overset{e_{21}}{\rightarrow} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \end{bmatrix} \tag{6}$$

In regard to this backward process,

$$\boldsymbol{P}_b = \boldsymbol{P}_{21} \cdot \boldsymbol{P}_{32} \cdot \boldsymbol{P}_{63} \tag{7}$$

Obviously, in this case, $\boldsymbol{P}_b \cdot \boldsymbol{P}_f = \boldsymbol{I}$ is obtained.

Figure 2 shows the graphical representation of the backward (from x_6 to x_1) solution based on Bellman-Ford algorithm. Note that some costs were modified in this figure because of the theoretical consistency.

3 Performance Preferability

Systems and Control Theory has been developed based on the principle of optimality. However, the optimality of control system may be an illusion of mathematician. The theorists of systems and control theory are very particular about continuous mathematics, i.e., differential and integral calculus, differental equation, and linear algebra, especially quadratic form (e.g., Riccati equation). However, what is 'continuous'?. We think that the technique of differentiation/integration and the concept of limitation and infinity (e.g.,$\lim_{\Delta x \to 0}$, $y \to \infty$) would not be necessary [6)]. Every computer-simulated systems are 'finite' and 'discrete'.

[6)] The author also thinks that the infinity dimensional space

3.1 The principle of optimality

R. Bellman proposed the following optimal policy with regard to the design of dynamical systems [7]. The principle of optimality is written below.

An optimal policy has the property that whatever the initial state and initial dicision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

However, now, we think that the important matters in the design of dynamic systems are the comparability and preferability of the performance. Our thoughts of this paper will be given later.

The dynamic programming for DEDSs was reconsideed in [14]. It is assumed there that the total cost is given the following additive form:

$$C(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N) \\ = \psi(\boldsymbol{x}_0, \boldsymbol{x}_1) + \psi(\boldsymbol{x}_1, \boldsymbol{x}_2) + \cdots + \psi(\boldsymbol{x}_{N-1}, \boldsymbol{x}_N), \tag{8}$$

where $\psi(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k)$, $k = 1, 2, \cdots, N$ are state-transition costs based on the following event sequences[7]:

$$\boldsymbol{\mathcal{E}}_N = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_N\}. \tag{9}$$

When considering dynamic systems as shown in (1) and (2), the minimization of costs will be given below:

$$\Psi_N(\boldsymbol{x}_N) = \min_{\boldsymbol{\mathcal{E}}_N} \; C(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N). \tag{10}$$

Thus, the following recurrrent equation is obtained:

$$\Psi_N(\boldsymbol{x}_N) = \min_{\boldsymbol{\mathcal{E}}_N} \; \left[\psi(\boldsymbol{x}_{N-1}, \boldsymbol{x}_N) + \Psi_{N-1}(\boldsymbol{x}_{N-1})\right] \tag{11}$$

based on the concept of *Dynamic Programing.*

3.2 Recurent inequalities

At present day, we are exactly in the digital computer society. Therefore, we think that the comparable viewpoints of system performance (i.e., preferability/inferiority) are rather important. For example, in regard to non-numerical (qualitative) problems, the following expression may be written:, e.g.,

$$\Psi_k(x_k) \preceq \psi(x_{k-1}, x_k) \cup \Psi_{k-1}(x_{k-1}),\ k = 1, 2, \cdots, N.$$

However, in any case the comparability of performance will be given by 'recurrence inequalities'. That

and moreover the function space, e.g., L_p, H_p, and really H_∞ spaces may be a 'dream world' for the theorists.

7) For example, e_{14}, e_{45}, e_{56} in (4) coorrespond to these event sequences.

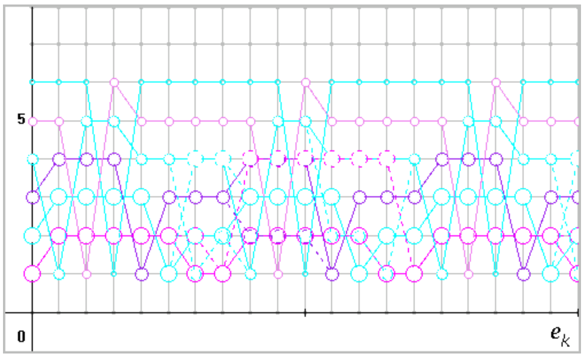

Fig. 3: Values of state-transition in each place-1.

is, they are nothing else reccurent 'if then ∼ else ∼' rules in the computer program. When using Pascal language, it can be written as:

---

if ('inequality' > or <) **then** ('substitution' :=)
else ('substitution' :=)

As is obvious, (11) can be written by the following recurrence inequalities,

$$\begin{cases} \Psi_1(\boldsymbol{x}_1) \leq \psi(\boldsymbol{x}_0, \boldsymbol{x}_1), & \forall\{\boldsymbol{e}_1\} \\ \Psi_2(\boldsymbol{x}_2) \leq \psi(\boldsymbol{x}_1, \boldsymbol{x}_2) + \Psi_1(\boldsymbol{x}_1), & \forall\{\boldsymbol{e}_1, \boldsymbol{e}_2\} \\ \quad \vdots \\ \Psi_{N-1}(\boldsymbol{x}_{N-1}) \leq \psi(\boldsymbol{x}_{N-2}, \boldsymbol{x}_{N-1}) + \Psi_{N-2}(\boldsymbol{x}_{N-2}), \\ \Psi_N(\boldsymbol{x}_N) \leq \psi(\boldsymbol{x}_{N-1}, \boldsymbol{x}_N) + \Psi_{N-1}(\boldsymbol{x}_{N-1}), \quad \forall \boldsymbol{\mathcal{E}}_{\boldsymbol{N}}. \end{cases} \tag{12}$$

Conversely, when $\boldsymbol{x}_N$ is fixed and $\Psi_N(\boldsymbol{x}_N) = \bar{\Psi}$ is assumed, the following (backward) recurrence inequalities can be obtained by subtracting $\bar{\Psi_N}$ in the both side of (12):

$$\begin{cases} \Psi^*_{N-1}(\boldsymbol{x}_{N-1}) \leq \psi(\boldsymbol{x}_{N-1}, \boldsymbol{x}_N), & \forall\{\boldsymbol{e}_N\}, \\ \Psi^*_{N-2}(\boldsymbol{x}_{N-2}) \leq \psi(\boldsymbol{x}_{N-2}, \boldsymbol{x}_{N-1}) + \Psi^*_{N-1}(\boldsymbol{x}_{N-1}), \\ \quad \vdots \\ \Psi^*_1(\boldsymbol{x}_1) \leq \psi(\boldsymbol{x}_1, \boldsymbol{x}_2) + \Psi^*_2(\boldsymbol{x}_2), \\ \Psi^*_0(\boldsymbol{x}_0) \leq \psi(\boldsymbol{x}_0, \boldsymbol{x}_1) + \Psi^*_1(\boldsymbol{x}_1), \quad \forall \boldsymbol{\mathcal{E}}_N \end{cases} \tag{13}$$

where $\Psi^*_k(\boldsymbol{x}_k) = \bar{\Psi}_N - \Psi_k(\boldsymbol{x}_k)$. Consequently, $\Psi^*_0(\boldsymbol{x}_0)$ will become equal to $\bar{\Psi}_N$.

3.3 Least-cost routing problem

Consider a cost-minimization (routing) problem as shown in Figs. 1 and 2, here. Bellman-Ford algorithm is simply written by the following recurrence

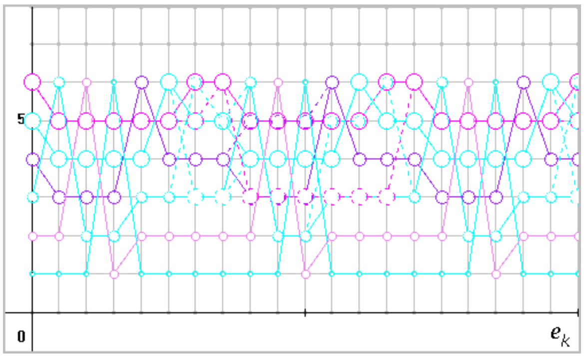

Fig. 4: Values of state-transitions in each place-2.

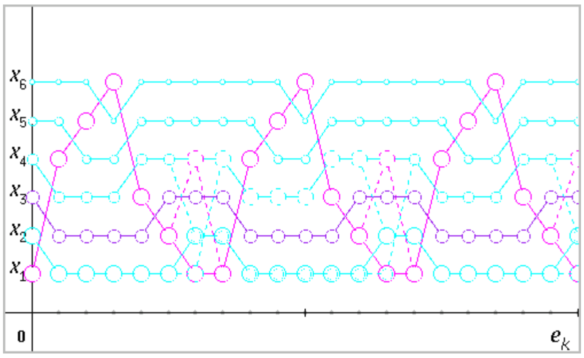

Fig. 5: State transitions with respect to each place.

inequalities [12]

$$L_{p+1}(x_j) \leq w(x_i, x_j) + L_p(x_i), \quad \forall i < j, \tag{14}$$

where

- $L_p(x_i)$: cost of the least-cost path from place x_1 to place x_i (for the backward case, from place x_6 to place x_i) under the constraint of no more than h links.
- $p = 1,2,3,4,5$(or $p = 6,7,8,9,10$): maximum number of links in a path.

Therefore, costs of the least-costs path are given as $L1(x_2) = 2$, $L_2(x_3) = 4 \rightarrow 3$, $L_3(x_4) = 1$, $L_4(x5) = 2$, $L_5(x_6) = 4$. For the backward case, $L_1(x_5) = 4$, $L_2(x_4) = 5$ $L_3(x_3) = 4$, $L_4(x_2) = 7 \rightarrow 3$, $L_5((x_1)$=5 are obtained. Thus, by using this algorithm, the reccurence process can be drawn as shown in Fig. 2.

Figure 3 shows the values of state-transition [8] $\hat{x}_i (i = 1, 3, \cdots, 6)$ in each place $x_j (j = 1, 2, \cdots, 6)$, when starting from

$$\hat{x}_1 = 1,\ \hat{x}_2 = 2,\ \hat{x}_3 = 3,\ \hat{x}_4 = 4,\ \hat{x}_5 = 5,\ \hat{x}_6 = 6.$$

[8] In the following, it is assumed that the event sequence,

$$\{e_0, e_1, e_2 \cdots, e_{N-1}\} = \{e(t_0), e(t_1), e(t_2) \cdots, e(t_{N-1})\}$$

is occurred at the same period as described in the classic control theory.

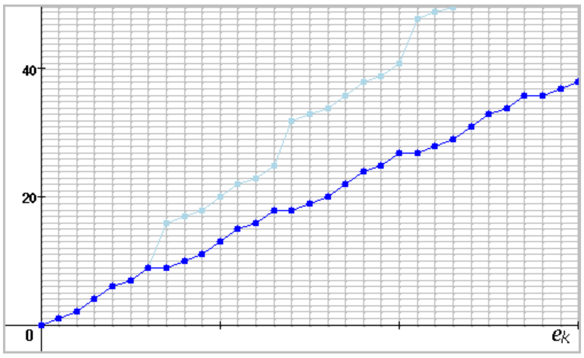

Fig. 6: Increasing total-costs of a DEDS with failure.

Figure 4 is the case where the initial state-value starts from

$$\hat{x}_1 = 6,\ \hat{x}_2 = 5,\ \hat{x}_3 = 4,\ \hat{x}_4 = 3,\ \hat{x}_5 = 2,\ \hat{x}_6 = 1.$$

On the other hand, Fig. 5 shows the process of each transition in these places. As is obvious, the latter is not dependent on the initial value of transitions.

In these figure, the dotted lines represent the case when a failure occurs to the place x_2 (from $x_2 \rightarrow x_1$) to x_2 to x_4. Figure 6 shows the inclеasing total-cost for the normal and failed cases. Here, the blue-line is normal case and the lightblue-line is failed case, respectively.

4 Security and Stability

In order to inspect the robust dynamic performance of DEDSs, we consider the following expression:

$$\boldsymbol{x}(t_{k+1}) = \boldsymbol{\mathcal{P}}(t_k)\boldsymbol{x}(t_k) + \boldsymbol{\mathcal{G}}(\boldsymbol{e}(t_k))\boldsymbol{x}(t_k), \tag{15}$$

where $\boldsymbol{\mathcal{P}}(\cdot)$ and $\boldsymbol{\mathcal{G}}(\cdot)$ are assumed to be piecewise constant matrices for $t_k \leq t < t_{k+1}$.

When corresponding the expression to (1), the second term of (15) is written as

$$\boldsymbol{\mathcal{G}}(\boldsymbol{e}(t_k))\boldsymbol{x}(t_k) = \boldsymbol{f}(\boldsymbol{x}(t_k), \boldsymbol{e}(t_k)) - \boldsymbol{\mathcal{P}}(t_k)\boldsymbol{x}(t_k). \tag{16}$$

When considering simply $\boldsymbol{\mathcal{G}}(t_k) := \boldsymbol{\mathcal{G}}(\boldsymbol{e}(t_k))$, each element is given by

$$g_{ij}(t_k) = \left(f_i(\boldsymbol{x}(t_k), \boldsymbol{e}(t_k)) - \sum_{l=1}^{n} p_{il}(t_k) x_l(t_k) \right) / x_j(t_k) \tag{17}$$

in regard to $\boldsymbol{\mathcal{G}} = [g_{ij}]$ and $\boldsymbol{\mathcal{P}} = [p_{ij}]$. Therefore, (15) can be expanded as follows:

$$\boldsymbol{x}(t_k) = \left(\prod_{h=0}^{k-1} \boldsymbol{\mathcal{P}}(t_h)\right) \boldsymbol{x}(t_0) + \sum_{l=1}^{k} \left(\prod_{h=l}^{k-1} \boldsymbol{\mathcal{P}}(t_h)\right) \boldsymbol{\mathcal{G}}(t_{l-1})\boldsymbol{x}(t_{l-1}). \quad (18)$$

4.1 Multiple metrics and boundedness

The metric in the state space (i.e., vector space) is usually defined by a scalar value. However, it may lead to a severe condition for the stability of some kind of nonlinear systems. Therefore, we consider the metric (i.e., norm) for each element of the state as follows:

$$\|x_i(t_k)\|_{\ell_\infty} := \sup_{1\le l\le k} |x_i(t_l)| \in \mathbb{Z}_+. \quad (19)$$

In this paper, we define simply as[9)].

$$\|x_i(t_k)\| \le |x_i(t_k)|, \quad \forall k, \quad x_i \in \mathbb{Z}_+ \quad (20)$$

When considering multiple metrics, the following vector can be written:

$$\|\boldsymbol{x}(t_k)\| := \begin{bmatrix} \|x_1(t_k)\| \\ \|x_2(t_k)\| \\ \vdots \\ \|x_n(t_K)\| \end{bmatrix} \in \mathbb{Z}_+^n. \quad (21)$$

Based on the above multiple vectors, the following simultaneous inequality can be obtained from (18) [10)]:

$$\|\boldsymbol{x}(t_k)\| \leqq \|\bar{\boldsymbol{x}}(t_K)\| + \left\|\sum_{l=1}^{k} \boldsymbol{\Phi}(k,l)\boldsymbol{x}(t_{l-1})\right\|. \quad (22)$$

Here, we define the following nonnegative matrix [16]:

$$\|\boldsymbol{\Theta}(t_k)\| := \begin{bmatrix} \|\theta_{11}(t_k)\| & \dots & \|\theta_{1n}(t_k)\| \\ \vdots & \ddots & \vdots \\ \|\theta_{n1}(t_k)\| & \dots & \|\theta_{nn}(t_k)\| \end{bmatrix},$$

where,

$$\|\theta_{ij}(t_k)\| = \left\|\sum_{l=1}^{k} \phi_{ij}(k,l)x_j(t_{l-1})\right\| / \|x_j(t_k)\| \quad i,j = 1,2,\cdots,n.$$

Therefore, inequality (22) is written as

$$\|\boldsymbol{x}(t_k)\| \leqq \|\bar{\boldsymbol{x}}(t_k)\| + \|\boldsymbol{\Theta}(t_k)\| \cdot \|x(t_k)\|. \quad (23)$$

9) In the following we write simply $\|x_i(t_k)\| := \|x_i(t_k)\|_{\ell_\infty}$.

10) Inequality symbols for matrices and vectors are based on [15]

Moreover, it can be given as

$$\Big(\boldsymbol{I} - \|\boldsymbol{\Theta}(t_k)\|\Big)\|\boldsymbol{x}(t_k)\|_{\ell_\infty} \leqq \|\bar{\boldsymbol{x}}(t_k)\|. \quad (24)$$

Here, we write the above inequality as follows:

$$(\boldsymbol{I} - \boldsymbol{C})\boldsymbol{X} \leqq \boldsymbol{Y}, \quad (25)$$

where $\boldsymbol{C} \geqq \boldsymbol{0}$, $\boldsymbol{X} \geqq \boldsymbol{0}$, and $\boldsymbol{Y} \geqq \boldsymbol{0}$ correspond to $\|\boldsymbol{\Theta}(t_k)\|$, $\|\boldsymbol{x}(t_k)\|$, and $\|\bar{\boldsymbol{x}}(t_k)\|$, respectively.

4.2 Numerical Considerations

Consider the following square matrix: [17]

$$\boldsymbol{A} = \boldsymbol{I} - \boldsymbol{C} = \begin{bmatrix} a_{11} & -a_{12} & \dots & -a_{1n} \\ -a_{21} & a_{22} & \dots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \dots & -a_{nn} \end{bmatrix} = \begin{bmatrix} 1-c_{11} & -c_{12} & \dots & -c_{1n} \\ -c_{21} & 1-c_{22} & \dots & -c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -c_{n1} & -c_{n2} & \dots & 1-c_{nn} \end{bmatrix}, \quad (26)$$

where $a_{ij} \ge 0$ $(i,j = 1,2,\dots,n)$. It is said that the most common situation in the biological, physical, and social sciences is where the matrix $\boldsymbol{A}$ has nonpositive off-diagonal and nonnegative diagonal entries [17]. Of course, it can simply be written as $\boldsymbol{A} = [a_{i,j}]$. However, in this paper, the expression of (26) will be used in order to clarify the sign of each element.

When $n = 6$, a simultaneous equation,

$$\begin{bmatrix} a_{11} & -a_{12} & \dots & -a_{16} \\ -a_{21} & a_{22} & \dots & -a_{26} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{51} & -a_{52} & \dots & -a_{56} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_6 \end{bmatrix} \leqq \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_6 \end{bmatrix}. \quad (27)$$

can be defined. By using an elimination method, the following equation is obtained based on an upper triangle matrix.

$$\begin{bmatrix} a_{11}^{(1)} & -a_{12}^{(1)} & \dots & -a_{16}^{(1)} \\ 0 & a_{22}^{(1)} & \dots & -a_{26}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -a_{66}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_6 \end{bmatrix} \leqq \begin{bmatrix} y_1^{(1)} \\ y_2^{(2)} \\ \vdots \\ y_6^{(6)} \end{bmatrix}, \quad (28)$$

where $a_{1j}^{(1)} = a_{1j}$, $(j = 1,2,\cdots,,6)$.

Furthermore,

$$\begin{cases} a_{ij}^{(2)} = \dfrac{1}{a_{11}^{(1)}} \begin{vmatrix} a_{11}^{(1)} & -a_{1j}^{(1)} \\ -a_{i1}^{(1)} & a_{ij}^{(1)} \end{vmatrix} \\ a_{ij}^{(3)} = -\dfrac{1}{a_{22}^{(2)}} \begin{vmatrix} a_{22}^{(2)} & -a_{2j}^{(2)} \\ -a_{31}^{(1)} & -a_{ij}^{(2)} \end{vmatrix}, \\ \vdots \\ a_{ij}^{(6)} = \dfrac{1}{a_{55}^{(5)}} \begin{vmatrix} a_{55}^{(5)} & -a_{5j}^{(5)} \\ -a_{i5}^{(5)} & a_{ij}^{(5)} \end{vmatrix}. \end{cases} \tag{29}$$

Therefore, the right side of (24) is given by

$$\begin{cases} y_1^{(1)} = y_1, \\ y_2^{(2)} = y_2^{(1)} + \dfrac{a_{21}^{(1)}}{a_{11}^{(1)}} y_1^{(1)}, \\ \vdots \\ y_6^{(6)} = y_6^{(5)} + \dfrac{a_{65}^{(5)}}{a_{55}^{(5)}} y_5^{(5)}. \end{cases} \tag{30}$$

and thus if $a_{11}^{(1)} > 0$, $a_{22}^{(2)} > 0$, $a_{33}^{(3)}$, $a_{44}^{(4)} > 0$, $a_{55}^{(5)} > 0$, and $a_{66}^{(6)} > 0$, the following inequalities can be obtained:

$$\begin{cases} x_6 \leq \dfrac{1}{a_{66}^{(6)}} y_6^{(6)} \\ x_5 \leq \dfrac{1}{a_{55}^{(5)}} \left(y_5^{(5)} + a_{56}^{(5)} x_6^{(5)} \right) \\ \vdots \\ x_1 \leq \dfrac{1}{a_{11}^{(1)}} \left(y_1^{(1)} + a_{12}^{(1)} x_2 + \cdots + a_{16}^{(1)} x_6 \right). \end{cases} \tag{31}$$

Thus, from (29) and (30), if the following inequalities,

$$a_{11}^{(1)} > 0,\ a_{22}^{(2)} > 0,\ a_{33}^{(3)},\ a_{44}^{(4)} > 0,\ a_{55}^{(5)} > 0,\ a_{66}^{(6)} > 0,$$

are satified, the security and stability will be guaranteed. [18, 20, 21]

5 Conclusion

In ihis paper, the analysis and design of DEDSs has been studied using state transferred vectors and permutation matrices. As an example of designing problem, a packet switching communication network (6th order system) was treated based on Bellman-Ford algorithm. And the optimality (preferability) of system performance was discussed based on the recurrent inequalities expression of dynamic programming.

However, we think that this paper may be insufficient with respect to the security and stability of DEDSs. The appropriate assertion and example will be shown in the conference, if possible. The above is not caused by 'COVID-19'.

References

[1] Y. Ho and X. Cao: *Perturbation Analysis of Disrete Event Dynamic Systems*, Kluwer Academic Publishers (1991)

[2] K. M. Passino and K. L. Burgess: *The Stability Analysis of Discrete Event Systems*, John Wiley & Sons., New York (1998)

[3] C. G. Cassandras and S. Lafortune: *Introduction to Discrete Event Systems*, Springer-Verlag, Berlin (1999)

[4] W. M. Wonham and K. Cai: *Supervisory Control of Discrete-Event Systems*, Springer International publishing AG (2019)

[5] K. You, N. Xiao, and L. Xie: *Analysis and Design of Networked Control Systems*, Springer-Verlag, London (2015)

[6] Y. B. Zhao, G. P. Liu, Y. Kang, and L. Yu: *Packet-Based Control for Networked Control Systems*, Science Press, Beijin, China (2017)

[7] R. Bellman: *Dynamic Programming*, Princeton University Presss, 1957, Dover Edition (2003)

[8] R. Bellman and S. Dreyfus: *Applied Dynamic Programming*, Princeton University Press (1962)

[9] R. B. Bapat and T. E. S. Raghavan: *Nonnegative Matrices and Applications*, Cambridge Univ. Press (1997)

[10] F. Zhang: *Matrix Theory: Basic Results and Techniques*, Springer-Verlag, New York, pp. 123-130 (1999)

[11] R. A. Horn and C. R. Johnson: *Matrix Analysis (2nd ed.)*, Cambridge Univ. Press (2013)

[12] W. Stallings: *Data and Computer Communications* (10th ed.) Pearson, UK (2014)

[13] L. R. Ford Jr. and D. R. Fulkerson, *Flows in Networks*, Princeton University Press (1962)

[14] Y. Okuyama: Dynamic Programming for Discrete-Event and Hybrid Systems, *Proc. of the SICE Annual Conference* (online) (2020) (to appear)

[15] J. M. Ortega: *Matrix Theory; A Second Course*, Plenum Press, New York, pp. 221-224 (1987)

[16] H. Minc: *Nonnegative Matrices*, J. Wiley & Sons, New York, pp. 68-104 (1988)

[17] A. Berman and R. J. Plemmons: *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, London, pp. 26-62 & pp. 132-164 (1979)

[18] A. M. Ostrowski: Note on Bounds for Some Determinants; *Duke Mathematical Journal*, Vol. 22, pp. 95-102 (1955)

[19] Y. Okuyama: On the L_2-Stability of Linear Systems with Time-Varying Parameters; *Transaction of SICE*, Vol. 3, pp. 252-259 (1967) (in Japanese).

[20] Y. Okuyama: *Discrete Control Systems*, Springer-Verlag, London, pp. 233-236 & 152-153 (2014)

[21] Y. Okuyama: Stability Analysis of Discrete Event Control Systems Based on Connection Matrices and Graphs, *Proc. of the 12th Asian Control Conference*, Kitakyushu, Japan, pp.255-260 (2019)