IEMS Vol. 3, No. 1, pp. 1-11, April 2004.

Derivation Algorithm of State-Space Equation for Production Systems Based on Max-Plus Algebra

Hiroyuki Goto†
Research and Consulting Division
The Japan Research Institute, Ltd., 16 Ichiban-cho, Chiyoda-ku, Tokyo 102-0082, JAPAN
Tel: +81-3-3288-4173, Fax: +81-3-3288-4338, E-mail: goto.hiroyuki@jri.co.jp

Shiro Masuda
Tokyo Metropolitan Institute of Technology, 6-6 Asahigaoka, Hino, Tokyo 191-0065, JAPAN
Tel & Fax: +81-42-585-8631, E-mail: smasuda@cc.tmit.ac.jp

Abstract. This paper proposes a new algorithm for determining an optimal control input for production systems. In many production systems, completion time should be planned within the due dates by taking into account precedence constraints and processing times. To solve this problem, the max-plus algebra is an effective approach. The max-plus algebra is an algebraic system in which the max operation is addition and the plus operation is multiplication, and similar operation rules to conventional algebra are followed. Utilizing the max-plus algebra, constraints of the system are expressed in an analogous way to the state-space description in modern control theory. Nevertheless, the formulation of a system is currently performed manually, which is very inefficient when applied to practical systems. Hence, in this paper, we propose a new algorithm for deriving a state-space description and determining an optimal control input with several constraint matrices and parameter vectors. Furthermore, the effectiveness of this proposed algorithm is verified through execution examples.

Keywords: model predictive control, production system, max-plus algebra, constraint matrix

1. INTRODUCTION

Precedence constraints or multi-stage processes are frequently imposed in many production systems such as assembly lines or batch-processing lines. They imply that a certain process cannot start until any former processes complete and all the manufactured parts for input are ready. In addition, completion times should be planned within the due dates. Under such conditions, the max-plus algebra is an effective approach for solving problems concerning the processing times and due dates (Boimond *et al*., 1996; Goto *et al*., 2002; Masuda *et al*., 2003).

The max-plus algebra is an algebraic system in which the max operation is addition and the plus operation is multiplication. Properties such as commutativity and the distributive law hold, both of which are familiar in conventional algebra. Any system taking into account both precedence constraints and processing times can be represented simply by utilizing the max-plus algebra. The system representation is called a MPL (max-plus-linear) system, which is similar to the state-space description in modern control theory (Baccelli *et al*., 1992; Cohen *et al*., 1989). An MPL system has an independent variable called the event counter, which is analogous to the use of “time” in a traditional representation.

An MPL system can describe the behavior of a TEG (Timed Event Graphs) (Cohen *et al*., 1989). The TEG is a subclass of Petri net in which all places have a single transition upstream and a single one downstream. Although the application scope for the TEG is limited to discrete event systems with only synchronization and no concurrency, they can be used to describe the behavior of many practical systems.

In the process industry, many applications utilizing the max-plus algebra have been studied and reported on recently. For instance, internal model control (IMC) is applied to the controller design method in Boimond *et al*. (1996), and a diagnosis method of batch processes based on IMC is proposed in Schullerus *et al*. (2001). In addition, model predictive control (MPC) is applied to a scheduling algorithm of assembly lines in Schutter *et al*.

† : Corresponding Author

(2001), and Goto *et al*. (2002) has extended this to cases in which the processing times differ by each event. As these few examples indicate, much attention has been paid to the application field of the max-plus algebra.

However, the related studies are still under investigation, and require improvement for general use in practical systems. For example, in production systems, precedence constraints are frequently changed and the state-space equations should be restructured accordingly. Currently, the equations have to be derived manually, which is a very inefficient process. Therefore, we propose a new algorithm for deriving a state-space equation and determining an optimal control input with only a few manual procedures. In this paper, we deal with deterministic flow-line style production systems with predetermined performance parameters.

Generally, a controller design utilizing MPC is constructed from the following steps:

- Model a control objective
- Examine the constraints and derive a state-space equation
- Derive an output prediction equation and determine a control law

The procedures described above would be laborsome under certain conditions. For instance, the derivation process of the state-space equations in step 2 is very time consuming for complicated systems. Even when an analogous system is to be analyzed, the design process must be started from step 1. When a robust controller should be designed, the prediction number must be large, and the prediction equation becomes complicated.

Hence, we propose a new algorithm to determine a system representation and an optimal control input. Utilizing the new algorithm, the state-space equation can be derived from only three constraint matrices and two parameter vectors. Accordingly, complicated systems can be analyzed easily.

This paper is organized as follows:

Section 2 gives the mathematical preliminaries. Section 3 shows a MPC framework for the MPL system using an example of a production system. Section 4 introduces an algorithm for deriving a state-space equation. Section 5 presents simulation results, and finally, section 6 gives concluding remarks.

2. MATHEMATICAL PRELIMINARIES

The max-plus algebra is an algebraic structure defined on $\boldsymbol{R}_\varepsilon = \boldsymbol{R} \cup \{-\infty\}$, where $\boldsymbol{R}$ represents the real field. The basic two operators $\oplus$ and $\otimes$, which stand for addition and multiplication respectively, are defined in the following way:

$$x \oplus y = \max\{x, y\}, \quad x \otimes y = x + y \tag{1}$$

Let ε be defined as $-\infty$, which is a unit element of the addition $\oplus$, and let e be defined as 0, which is a unit element of the multiplication $\otimes$. These operators have similar operation rules like the conventional $(+, \times)$ algebra; $\otimes$ operator has a higher priority than $\oplus$ operator, and they hold the distributive law:

$$(x \oplus y) \otimes z = x \otimes z \oplus y \otimes z \tag{2}$$

For simplicity, the $\otimes$ operator is often suppressed. Moreover, the following two operators are also defined:

$$x \wedge y = \min(x, y), \quad x \setminus y = -x + y \tag{3}$$

Operations to multiple numbers are defined in the followings. When $m \le n$, then

$$\bigoplus_{k=m}^{n} a_k = a_m \oplus a_{m+1} \oplus \cdots \oplus a_n = \max(a_m, a_{m+1}, \cdots, a_n) \tag{4}$$

$$\bigotimes_{k=m}^{n} a_k = a_m \otimes a_{m+1} \otimes \cdots \otimes a_n = a_m + a_{m+1} + \cdots + a_n \tag{5}$$

$$\bigwedge_{k=m}^{n} a_k = a_m \wedge a_{m+1} \wedge \cdots \wedge a_n = \min(a_m, a_{m+1}, \cdots, a_n) \tag{6}$$

Furthermore, operation rules on matrices are defined by applying the same law described above. For instance, in $\boldsymbol{A}, \boldsymbol{B} \in \boldsymbol{R}_\varepsilon^{m \times n}$,

$$[\boldsymbol{A} \oplus \boldsymbol{B}]_{ij} = [\boldsymbol{A}]_{ij} \oplus [\boldsymbol{B}]_{ij} = \max([\boldsymbol{A}]_{ij}, [\boldsymbol{B}]_{ij}) \tag{7}$$

$$[\boldsymbol{A} \wedge \boldsymbol{B}]_{ij} = [\boldsymbol{A}]_{ij} \wedge [\boldsymbol{B}]_{ij} = \min([\boldsymbol{A}]_{ij}, [\boldsymbol{B}]_{ij}) \tag{8}$$

where $[*]_{ij}$ expresses (i,j)-th element of the matrix. Additionally, $[*]_{i\cdot}$ denotes i-th row of the matrix.

If $\boldsymbol{A} \in \boldsymbol{R}_\varepsilon^{m \times l}$, $\boldsymbol{B} \in \boldsymbol{R}_\varepsilon^{l \times p}$,

$$[\boldsymbol{A} \otimes \boldsymbol{B}]_{ij} = \bigoplus_{k=1}^{l} ([\boldsymbol{A}]_{ik} \otimes [\boldsymbol{B}]_{kj}) = \max_{k=1,\cdots,l} ([\boldsymbol{A}]_{ik} + [\boldsymbol{B}]_{kj}) \tag{9}$$

$$[\boldsymbol{A} \Theta \boldsymbol{B}]_{ij} = \bigwedge_{k=1}^{l} ([\boldsymbol{A}]_{ik} \setminus [\boldsymbol{B}]_{kj}) = \min_{k=1,\cdots,l} (-[\boldsymbol{A}]_{ik} + [\boldsymbol{B}]_{kj}) \tag{10}$$

Unit elements of addition and multiplication are stated as follows:

$\boldsymbol{\varepsilon}_{mn}$: All elements are ε in $\boldsymbol{\varepsilon}_{mn} \in \boldsymbol{R}_\varepsilon^{m\times n}$

$\boldsymbol{e}_m$: Only diagonal elements are e and all off-diagonal elements are ε in $\boldsymbol{e}_m \in \boldsymbol{R}_\varepsilon^{m\times m}$

If $\boldsymbol{a}, \boldsymbol{b} \in \boldsymbol{R}_\varepsilon^{n}$, $\boldsymbol{a} \le \boldsymbol{b}$ implies $[\boldsymbol{a}]_i \le [\boldsymbol{b}]_i$ for all $(1 \le i \le n)$.

3. DETERMINATION OF OPTIMAL CONTROL INPUT

This section introduces a MPL system using the model of a production system and considers the optimal control law for the system. Firstly, we briefly review the design method for the model predictive control based on MPL systems described in Goto *et al*. (2002) in order to show the effectiveness of the modeling of MPL systems.

3.1 State-Space Description Using Max-Plus Algebra

Using the max-plus algebra, constraints on production systems can be expressed in a form that is similar to the state-space representation in modern control theory.

Figure 1 shows a two-input, one-output production system. Machine 1 receives parts from input lane 1, processes them, and sends them to machine 3. Machine 2 receives parts from input lane 2, processes them, and sends them to machine 3. Machine 3 receives two parts from machines 1 and 2, processes them, and sends the resulting part to the next lane. Suppose each machine can process just a single item at a time, and the inventory buffer sizes are infinite.

Let variables relevant to the production system be defined as follows:

1) $u_1(k)$ $u_2(k)$:
Input times for the k-th part on machines 1 and 2
2) $d_1(k)$, $d_2(k)$, $d_3(k)$:
Processing times for the k-th part on machines 1, 2 and 3
3) $y(k)$:
Completion time of processing for the k-th part on machine 3
4) $x_1(k)$, $x_2(k)$, $x_3(k)$:
Starting times of processing for the k-th part on machines 1, 2 and 3

Suppose the following precedence constraints are imposed on the production system.

- Machines 1 and 2 cannot start processing until any previous parts have completed processing and the next parts have been inputted.
- Machine 3 cannot start processing until any previous parts have completed processing and the next parts are received from machines 1 and 2.
- Each machine starts processing as soon as all necessary parts are available.

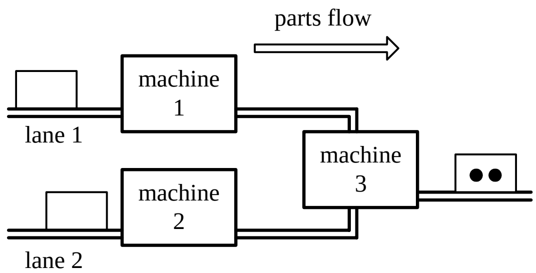

Figure 1. Two-input, one-output production system

In terms of the starting times for the processing, they are formulated as

$$x_1(k+1) = \max\{x_1(k) + d_1(k),\ u_1(k+1)\} \tag{11}$$
$$x_2(k+1) = \max\{x_2(k) + d_2(k),\ u_2(k+1)\} \tag{12}$$
$$x_3(k+1) = \max\{x_3(k) + d_3(k),\ x_1(k+1) + d_1(k+1),\ x_2(k+1) + d_2(k+1)\} \tag{13}$$

for $k \ge 1$, and set $x_1(0) = x_2(0) = x_3(0) = \varepsilon$ for the initial conditions. On the other hand, the completion time is formulated in the following way:

$$y(k) = x_3(k) + d_3(k) \tag{14}$$

Since Eqs. (11) ~ (14) are expressed only by max and + operators, they are replaced by $\oplus$ and $\otimes$ respectively. By substituting Eqs. (11)) ~ (12) into (13), they are reduced to the following linear representations.

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\,\boldsymbol{x}(k) \oplus \boldsymbol{B}\,\boldsymbol{u}(k+1) \tag{15}$$
$$\boldsymbol{y}(k) = \boldsymbol{C}\,\boldsymbol{x}(k) \tag{16}$$

where

$$\boldsymbol{A} = \begin{bmatrix} d_1(k) & \varepsilon & \varepsilon \\ \varepsilon & d_2(k) & \varepsilon \\ d_1(k)d_1(k+1) & d_2(k)d_2(k+1) & d_3(k) \end{bmatrix} \tag{17}$$

$$\boldsymbol{B} = \begin{bmatrix} e & \varepsilon \\ \varepsilon & e \\ d_1(k+1) & d_2(k+1) \end{bmatrix} \tag{18}$$

$$\boldsymbol{C} = [\varepsilon \quad \varepsilon \quad d_3(k)] \tag{19}$$
$$\mathbf{x}(k) = [x_1(k) \quad x_2(k) \quad x_3(k)]^T \tag{20}$$
$$\mathbf{y}(k) = [y(k)]^T \tag{21}$$
$$\mathbf{u}(k) = [u_1(k) \quad u_2(k)]^T \tag{22}$$

Thus, the precedence constraints on the production system depicted in Figure 1 can be expressed as linear equations in the max-plus algebra, which is similar to the state-space equations in modern control theory. Generally, systems whose constraints are represented by Eqs. (15)~(16), are called max-plus-linear (MPL) systems.

3.2 MPL System and Output Prediction Equation

In Eqs. (17)) ~ (19), $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ include processing times (system parameters) in their elements, and they depend on the event counter k. Thus, the matrices are also dependent upon the event counter, and we denote them as $\boldsymbol{A}_{\underline{k}}$, $\boldsymbol{B}_{\underline{k}}$ and $\boldsymbol{C}_{\underline{k}}$, respectively. Using these notations, the generalized linear state-space description in the max-plus algebra is expressed in the following way:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}_{\underline{k}}\, \boldsymbol{x}(k) \oplus \boldsymbol{B}_{\underline{k}}\, \boldsymbol{u}(k+1) \tag{23}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}_{\underline{k}}\, \boldsymbol{x}(k) \tag{24}$$

where

$$\boldsymbol{x}(k) \in \boldsymbol{R}_{\varepsilon}^{\ n} \quad , \quad \boldsymbol{u}(k) \in \boldsymbol{R}_{\varepsilon}^{\ p} \quad , \quad \boldsymbol{y}(k) \in \boldsymbol{R}_{\varepsilon}^{\ q}$$

$$\boldsymbol{A}_{\underline{k}} \in \boldsymbol{R}_{\varepsilon}^{\ n\times n} \quad , \quad \boldsymbol{B}_{\underline{k}} \in \boldsymbol{R}_{\varepsilon}^{\ n\times p} \quad , \quad \boldsymbol{C}_{\underline{k}} \in \boldsymbol{R}_{\varepsilon}^{\ q\times n}$$

$\boldsymbol{x}(k)$, $\boldsymbol{u}(k)$ and $\boldsymbol{y}(k)$ are called state variables, input variables, and output variables, respectively. n, p and q are the number of state variables, the number of inputs, and the number of outputs, respectively.

By utilizing Eq. (23) iteratively, the following equations are obtained.

$$\begin{cases} \boldsymbol{x}(k+1) &= \boldsymbol{A}_{\underline{k}}\, \boldsymbol{x}(k) \oplus \boldsymbol{B}_{\underline{k}}\, \boldsymbol{u}(k+1) \\ \boldsymbol{x}(k+2) &= \boldsymbol{A}_{\underline{k+1}}\boldsymbol{A}_{\underline{k}}\, \boldsymbol{x}(k) \\ & \quad \oplus \boldsymbol{A}_{\underline{k+1}}\boldsymbol{B}_{\underline{k}}\, \boldsymbol{u}(k+1) \oplus \boldsymbol{B}_{\underline{k+1}}\boldsymbol{u}(k+2) \\ \quad \vdots & \\ \boldsymbol{x}(k+N) &= \boldsymbol{A}_{\underline{k+N-1}} \cdots \boldsymbol{A}_{\underline{k}}\, \boldsymbol{x}(k) \\ & \quad \oplus \boldsymbol{A}_{\underline{k+N-1}} \cdots \boldsymbol{B}_{\underline{k}}\, \boldsymbol{u}(k+1) \oplus \cdots \\ & \quad \cdots \oplus \boldsymbol{B}_{\underline{k+N-1}}\, \boldsymbol{u}(k+N) \end{cases} \tag{25}$$

where N represents the prediction step number. Multiplying both sides of Eq. (25) by $\boldsymbol{C}_{\underline{k+1}}, \cdots, \boldsymbol{C}_{\underline{k+N}}$, and utilizing Eq. (24), we obtain the equation below.

$$\boldsymbol{Y}(k+1) = \boldsymbol{\Gamma}_{\underline{k}}\, \boldsymbol{x}(k) \oplus \boldsymbol{\Delta}_{\underline{k}}\, \boldsymbol{U}(k+1) \tag{26}$$

where

$$\boldsymbol{Y}(k+1) = \begin{bmatrix} \boldsymbol{y}(k+1) \\ \boldsymbol{y}(k+2) \\ \vdots \\ \boldsymbol{y}(k+N) \end{bmatrix} \quad , \quad \boldsymbol{U}(k+1) = \begin{bmatrix} \boldsymbol{u}(k+1) \\ \boldsymbol{u}(k+2) \\ \vdots \\ \boldsymbol{u}(k+N) \end{bmatrix} \tag{27}$$

$$\boldsymbol{\Gamma}_{\underline{k}} = \begin{bmatrix} \boldsymbol{C}_{\underline{k+1}}\boldsymbol{A}_{\underline{k}} \\ \boldsymbol{C}_{\underline{k+2}}\boldsymbol{A}_{\underline{k+1}}\boldsymbol{A}_{\underline{k}} \\ \vdots \\ \boldsymbol{C}_{\underline{k+N}}\boldsymbol{A}_{\underline{k+N-1}} \cdots \boldsymbol{A}_{\underline{k}} \end{bmatrix} \tag{28}$$

$$\boldsymbol{\Delta}_{\underline{k}} = \begin{bmatrix} \boldsymbol{C}_{\underline{k+1}}\boldsymbol{B}_{\underline{k}} & \cdots & \boldsymbol{\varepsilon}_{qp} \\ \boldsymbol{C}_{\underline{k+2}}\boldsymbol{A}_{\underline{k+1}}\boldsymbol{B}_{\underline{k}} & & \vdots \\ \vdots & \ddots & \boldsymbol{\varepsilon}_{qp} \\ \boldsymbol{C}_{\underline{k+N}}\boldsymbol{A}_{\underline{k+N-1}} \cdots \boldsymbol{A}_{\underline{k+1}}\boldsymbol{B}_{\underline{k}} & \cdots & \boldsymbol{C}_{\underline{k+N}}\boldsymbol{B}_{\underline{k+N-1}} \end{bmatrix} \tag{29}$$

Eq. (26) represents an output prediction equation, and it can be utilized to determine an optimal control input to the system.

3.3 Optimal Control Input

This subsection gives an optimal control law for MPL systems by using the output prediction equation.

Suppose the due dates are given by $r_i(k+j) \in \boldsymbol{R}_{\varepsilon}$ ($1 \leq i \leq q$, $1 \leq j \leq N$), which are also called reference signals. After specifying the reference signals, an optimal control input to the system is determined by solving

$$\boldsymbol{R}(k+1) = \boldsymbol{\Gamma}_{\underline{k}}\, \boldsymbol{x}(k) \oplus \boldsymbol{\Delta}_{\underline{k}}\, \boldsymbol{U}(k+1) \tag{30}$$

for $\boldsymbol{U}(k+1)$, where

$$\boldsymbol{R}(k+1) = \begin{bmatrix} \boldsymbol{r}(k+1) \\ \boldsymbol{r}(k+2) \\ \vdots \\ \boldsymbol{r}(k+N) \end{bmatrix} \quad , \quad \boldsymbol{r}(k+i) = \begin{bmatrix} r_1(k+i) \\ r_2(k+i) \\ \vdots \\ r_q(k+i) \end{bmatrix} \tag{31}$$

However, the solution of Eq. (30) cannot be obtained directly unlike in conventional $(+,\times)$ algebra. Consequently, in this paper, it is solved by utilizing the greatest subsolution method described in Cohen *et al.* (1989) after transforming Eq. (30).

Firstly, let Eq. (30) be transformed into

$$\boldsymbol{\Delta}_{\underline{k}}\, \boldsymbol{U}(k+1) = \boldsymbol{R}(k+1) \oplus \boldsymbol{\Gamma}_{\underline{k}}\, \boldsymbol{x}(k) \ , \tag{32}$$

which is justified by Cohen *et al.* (1989). Eq. (32) has the form of a linear equation in the max-plus algebra, which implies that getting the desired input using Eq. (30) reduces to solving the linear equation.

If $\boldsymbol{M} \in \boldsymbol{R}_{\varepsilon}^{m\times n}, \boldsymbol{z} \in \boldsymbol{R}_{\varepsilon}^{n}, \boldsymbol{v} \in \boldsymbol{R}_{\varepsilon}^{n}$, the greatest subsolution $\bar{\boldsymbol{z}}$ of a linear equation

$$\boldsymbol{M}\boldsymbol{z} = \boldsymbol{v} \tag{33}$$

is given by

$$\bar{\boldsymbol{z}} = \boldsymbol{M}^{T} \Theta\, \boldsymbol{v} \tag{34}$$

(Cohen *et el*., 1989). Therefore, the solution of Eq. (32) which gives the optimal control input, is expressed as follows:

$$\boldsymbol{U}(k+1) = \boldsymbol{\Delta}_{\underline{k}}{}^{T} \Theta \left\{ \boldsymbol{R}(k+1) \oplus \boldsymbol{\Gamma}_{\underline{k}}\, \boldsymbol{x}(k) \right\} \tag{35}$$

The following properties hold for the solution given in Eq. (35); when the processing can be finished precisely at the desired time, the solution represents the corresponding input time, and when it is not possible for the processing to complete on time, it gives the latest time at which the processing can be completed within the due date. Furthermore, if delayed completion is inevitable, the time that is given is the earliest time with a minimum delay.

The inputs to the system are determined by utilizing the Receding Horizon method, which gives only the $(k+1)$-th input whereas the inputs for $(k+1,\cdots,k+N)$ are obtained in Eq. (35). Specifically, they are calculated in the following way:

$$\boldsymbol{u}(k+1) = [\boldsymbol{e}_p \quad \boldsymbol{\varepsilon}_p \quad \cdots \quad \boldsymbol{\varepsilon}_p]\, \boldsymbol{U}(k+1) \tag{36}$$

The inputs after the $(k+1)$-th step are determined by recalculating Eq. (36) as the event counter increases. Thus, a feedback control against changes of the internal state can be realized.

4. INTERNAL REPRESENTATIONS

In the previous section, derivation processes of the state-space equation and the optimal control input were introduced. However, since they are performed manually in current research, they should be automated for practical use. Therefore, this section proposes a new algorithm for deriving the state-space equation and for determining an optimal control input by utilizing the max-plus algebra.

4.1 gives constraint matrices which specify the constraints of the production systems. 4.2 introduces an algorithm for deriving the state-space equation. 4.3 proposes an internal representation of the system matrices, and 0 derives an internal representation of the optimal control input.

4.1 Constraint Matrices

Precedence constraints of the system are specified by constraint matrices whose elements are logical numbers.

Let $\boldsymbol{L}$ denote the logical field. There are three constraint matrices which are defined as follows:

$PU \in \boldsymbol{L}^{n\times p}$: specifies the positions of input

$$[PU]_{ij} = \begin{cases} 1: \text{if machine } i \text{ is attached to the } j\text{-th input} \\ 0: \text{otherwise} \end{cases} \tag{37}$$

$PX \in \boldsymbol{L}^{n\times n}$: specifies the precedence constraints

$$[PX]_{ij} = \begin{cases} 1: \text{if machine } i \text{ receives processed parts from machine } j \\ 0: \text{otherwise} \end{cases} \tag{38}$$

$PY \in \boldsymbol{L}^{q\times n}$: specifies the positions of output

$$[PY]_{ij} = \begin{cases} 1: \text{if machine } j \text{ is attached to the } i\text{-th output} \\ 0: \text{otherwise} \end{cases} \tag{39}$$

Let us demonstrate the contents of the constraint matrices for the system depicted in Figure 1. Firstly, the input constraints are to be considered. The inputs 1 and 2 are connected to machines 1 and 2 respectively, which results in

$$PU = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{40}$$

Next, we consider the precedence constraints. In all machines of the system, only machine 3 has the constraint of machines 1 and 2. Therefore, PX is represented as

$$PX = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \tag{41}$$

As for the output constraint, output 1 is connected to machine 3. Hence, PY is expressed as

$$PY = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{42}$$

4.2 Derivation of State-Space Equation

This subsection derives the state-space representation from the constraint matrices. Consider the precedence constraints for a general system in an analogous way to the system shown in Figure 1. As for machine i, processing for the k+1-th part starts just after the following conditions are wholly satisfied:

- The processing for the *k*-th part is completed in the corresponding machine; the finishing time is given by $d_i(k) \otimes x_i(k)$.
- If the machine receives parts from an upstream machine *j*, the *k*+1-th part is inputted from machine *j*; the criterion is active when $[PX]_{ij} = 1$, and the input time is given by $d_j(k+1) \otimes x_j(k+1)$.

- If the machine is attached to the j-th input, the k+1-th part is inputted by the j-th input; the criterion is active when $[PU]_{ij}=1$, and the input time is given by $u_j(k+1)$.

Furthermore, the following condition holds for the output time.

- If the i-th output is attached to machine j, the completion time for the k-th part is given by $d_j(k)\otimes x_j(k)$; the criterion is active when $[PY]_{ij}=1$.

These are summarized in the following way:

$$\boldsymbol{x}(k+1)=\boldsymbol{A}_{\underline{k}}^{(0)}\,\boldsymbol{x}(k)\oplus \boldsymbol{F}_{\underline{k}}^{(0)}\boldsymbol{x}(k+1)\oplus \boldsymbol{B}_{\underline{k}}^{(0)}\,\boldsymbol{u}(k+1) \tag{43}$$

$$\boldsymbol{y}(k)=\boldsymbol{C}_{\underline{k}}\,\boldsymbol{x}(k) \tag{44}$$

where

$$\left[\boldsymbol{A}_{\underline{k}}^{(0)}\right]_{ij}=\begin{cases} d_i(k)\ :\text{if}\ \ i=j \\ \varepsilon : \text{otherwise}\end{cases} \tag{45}$$

$$\left[\boldsymbol{F}_{\underline{k}}^{(0)}\right]_{ij}=\begin{cases} d_j(k+1)\ :\text{if}\ \ [PX]_{ij}=1 \\ \varepsilon : \text{otherwise}\end{cases} \tag{46}$$

$$\left[\boldsymbol{B}_{\underline{k}}^{(0)}\right]_{ij}=\begin{cases} e : \text{if}\ \ [PU]_{ij}=1 \\ \varepsilon : \text{otherwise}\end{cases} \tag{47}$$

$$\left[\mathbf{C}_{\underline{k}}\right]_{ij}=\begin{cases} d_i(k)\ :\text{if}\ \ [PY]_{ij}=1 \\ \varepsilon : \text{otherwise}\end{cases} \tag{48}$$

Next, we transform Eq. (43) into the form of Eq. (15) by eliminating $\boldsymbol{x}(k+1)$ from the right hand side. In Eq. (43), when all elements of the i-th row in $\boldsymbol{F}_{\underline{k}}^{(0)}$ are ε, the variable $x_i(k+1)$ is said to be 'eliminated', and is said to be 'un-eliminated' otherwise. The transformation is performed by repeating the substitution of eliminated variables into un-eliminated variables.

Let $x_{\omega_1}(k+1), x_{\omega_2}(k+1), \cdots x_{\omega_g}(k+1)$ be eliminated variables, and let $x_{\psi_1}(k+1), x_{\psi_2}(k+1), \cdots x_{\psi_h}(k+1)$ be un-eliminated variables, where

$$\begin{aligned}&\boldsymbol{\omega}=\{\omega_i\in \boldsymbol{Z}, 1\le i\le g\} \\ &\boldsymbol{\psi}=\{\psi_j\in \boldsymbol{Z}, 1\le j\le h\}\end{aligned}\ ,\quad \begin{aligned}&\boldsymbol{\omega}\cap\boldsymbol{\psi}=\phi,\ \boldsymbol{\omega}\cup\boldsymbol{\psi}=\boldsymbol{Z} \\ &\boldsymbol{Z}=\{1,2,\cdots n\}\end{aligned} \tag{49}$$

Taking Eq. (43) into account, these variables are expressed as:

$$x_{\omega_i}(k+1)=\left[\mathbf{A}_{\underline{k}}^{(w)}\right]_{\omega_i:}\mathbf{x}(k)\oplus\left[\mathbf{B}_{\underline{k}}^{(w)}\right]_{\omega_i:}\mathbf{u}(k+1)\ ,\quad (1\le i\le g) \tag{50}$$

$$\begin{aligned}x_{\psi_j}(k+1)=&\left[\mathbf{A}_{\underline{k}}^{(w)}\right]_{\psi_j:}\mathbf{x}(k)\oplus\left[\mathbf{F}_{\underline{k}}^{(w)}\right]_{\psi_j:}\mathbf{x}(k+1) \\ &\oplus\left[\mathbf{B}_{\underline{k}}^{(w)}\right]_{\psi_j:}\mathbf{u}(k+1)\end{aligned}\ ,\quad (1\le j\le h), \tag{51}$$

Table 1. Transformation of the state variables

loop {
  Divide the state variables: {
    $x_{\omega_i}(k+1)\ (1\le i\le g)$: eliminated
    $x_{\psi_j}(k+1)\ (1\le j\le h)$: un-eliminated
  }
  if ($h=0$) exit loop
  loop for $\psi_j\ (1\le j\le h)$ {
    loop for $\omega_i\ (1\le i\le g)$ {
      if ($x_{\psi_j}(k+1)$ is un-eliminated for $x_{\omega_i}(k+1)$){
        substitute $x_{\omega_i}(k+1)$ for $x_{\psi_j}(k+1)$
      }
    }
  }
}

where w holds a non-negative integer number and is incremented at every substitution. Substituting Eq. (50) for Eq. (51), the following equation is obtained.

$$\begin{aligned}
&x_{\psi_j}(k+1)\\
&=\left[\mathbf{A}_{\underline{k}}^{(w)}\right]_{\psi_j:}\mathbf{x}(k)\\
&\quad\oplus\bigoplus_{\substack{l=1,\\ l\ne\omega_i}}^{n}\left\{\left[\mathbf{F}_{\underline{k}}^{(w)}\right]_{\psi_j l}x_l(k+1)\right\}\oplus\left[\mathbf{F}_{\underline{k}}^{(w)}\right]_{\psi_j\omega_i}x_{\omega_i}(k+1)\\
&\quad\oplus\left[\mathbf{B}_{\underline{k}}^{(w)}\right]_{\psi_j:}\mathbf{u}(k+1)\\
&=\left[\mathbf{A}_{\underline{k}}^{(w)}\right]_{\psi_j:}\mathbf{x}(k)\\
&\quad\oplus\bigoplus_{\substack{l=1,\\ l\ne\omega_i}}^{n}\left\{\left[\mathbf{F}_{\underline{k}}^{(w)}\right]_{\psi_j l}x_l(k+1)\right\}\oplus\varepsilon x_{\omega_i}(k+1)\\
&\quad\oplus\left[\mathbf{F}_{\underline{k}}^{(w)}\right]_{\psi_j\omega_i}\left\{\left[\mathbf{A}_{\underline{k}}^{(w)}\right]_{\omega_i:}\mathbf{x}(k)\oplus\left[\mathbf{B}_{\underline{k}}^{(w)}\right]_{\omega_i:}\mathbf{u}(k+1)\right\}\\
&\quad\oplus\left[\mathbf{B}_{\underline{k}}^{(w)}\right]_{\psi_j:}\mathbf{u}(k+1)\\
&=\left[\mathbf{A}_{\underline{k}}^{(w+1)}\right]_{\psi_j:}\mathbf{x}(k)\\
&\quad\oplus\left[\mathbf{F}_{\underline{k}}^{(w+1)}\right]_{\psi_j:}\mathbf{x}(k+1)\oplus\left[\mathbf{B}_{\underline{k}}^{(w+1)}\right]_{\psi_j:}\mathbf{u}(k+1)
\end{aligned} \tag{52}$$

where

$$\left[\mathbf{A}_{\underline{k}}^{(w+1)}\right]_{\psi_j:}=\left[\mathbf{A}_{\underline{k}}^{(w)}\right]_{\psi_j:}\oplus\left[\mathbf{F}_{\underline{k}}^{(w)}\right]_{\psi_j\omega_i}\left[\mathbf{A}_{\underline{k}}^{(w)}\right]_{\omega_i:} \tag{53}$$

$$\left[\mathbf{F}_{\underline{k}}^{(w+1)}\right]_{\psi_j l}=\begin{cases}\left[\mathbf{F}_{\underline{k}}^{(w)}\right]_{\psi_j l} & (l\ne\omega_i)\\ \varepsilon & (l=\omega_i)\end{cases}\ ,\quad (1\le l\le n) \tag{54}$$

$$\left[\mathbf{B}_{\underline{k}}^{(w+1)}\right]_{\psi_j:}=\left[\mathbf{B}_{\underline{k}}^{(w)}\right]_{\psi_j:}\oplus\left[\mathbf{F}_{\underline{k}}^{(w)}\right]_{\psi_j\omega_i}\left[\mathbf{B}_{\underline{k}}^{(w)}\right]_{\omega_i:} \tag{55}$$

On the other hand, since $x_i(k+1)\ (i\ne\psi_j)$ is invariant through this substitution, $[\boldsymbol{A}_{\underline{k}}^{(w)}]_{i:}$, $[\boldsymbol{F}_{\underline{k}}^{(w)}]_{i:}$, and

$[\boldsymbol{B}_{\underline{k}}^{(w)}]_{i:}\quad (i \neq \psi_j)$ are also invariant. Hence, define

$$[\boldsymbol{A}_{\underline{k}}^{(w+1)}]_{i:} = [\boldsymbol{A}_{\underline{k}}^{(w)}]_{i:}\ ,\quad [\boldsymbol{F}_{\underline{k}}^{(w+1)}]_{i:} = [\boldsymbol{F}_{\underline{k}}^{(w)}]_{i:}\ ,$$
$$[\boldsymbol{B}_{\underline{k}}^{(w+1)}]_{i:} = [\boldsymbol{B}_{\underline{k}}^{(w)}]_{i:}\quad \text{for all}\quad (1 \le i \le n, i \neq \psi_j) \tag{56}$$

Consequently, $[\boldsymbol{F}_{\underline{k}}^{(w)}]_{\psi_j \omega_i}$ turns into ε through this substitution. In other words, this operation is effective only when $[\boldsymbol{F}_{\underline{k}}^{(w)}]_{\psi_j \omega_i} \neq \varepsilon$. Let $x_{\psi_j}(k+1)$ be referred to as 'eliminated for $x_{\omega_i}(k+1)$ ' when $[\boldsymbol{F}_{\underline{k}}^{(w)}]_{\psi_j \omega_i} = \varepsilon$, and 'un-eliminated for $x_{\omega_i}(k+1)$ ' otherwise. The transformation from Eq. (43) into Eq. (23) is performed by the procedure shown in Table 1.

As an example, let us transform the state-space equation of the system shown in Figure 1. In accordance with Eqs. (45)) ~ (48), the system whose constraint matrices are given by Eqs. (40)) ~ (42) is initially represented by the following system matrices:

$$\left[\boldsymbol{A}_{\underline{k}}^{(0)}\right] = \begin{bmatrix} d_1(k) & \varepsilon & \varepsilon \\ \varepsilon & d_2(k) & \varepsilon \\ \varepsilon & \varepsilon & d_3(k) \end{bmatrix}$$

$$\left[\boldsymbol{F}_{\underline{k}}^{(0)}\right] = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ d_1(k+1) & d_2(k+1) & \varepsilon \end{bmatrix}$$

$$\left[\boldsymbol{B}_{\underline{k}}^{(0)}\right] = \begin{bmatrix} e & \varepsilon \\ \varepsilon & e \\ \varepsilon & \varepsilon \end{bmatrix}$$

$$[\boldsymbol{C}_{\underline{k}}] = [\varepsilon \quad \varepsilon \quad d_3(k)]$$

Checking each row of $[\boldsymbol{F}_{\underline{k}}^{(0)}]$ reveals that $x_1(k+1)$ and $x_2(k+1)$ are the eliminated variables, and $x_3(k+1)$ is the un-eliminated variable. Hence, $\boldsymbol{\omega} = \{1,2\}\ (g=2)$ and $\boldsymbol{\psi} = \{3\}\ (h=1)$. Utilizing Eqs. (53)) ~ (55), substituting $x_1(k+1)$ for $x_3(k+1)$ produces the following results:

$$\begin{aligned}\left[\mathbf{A}_{\underline{k}}^{(1)}\right]_{3:} &= \left[\mathbf{A}_{\underline{k}}^{(0)}\right]_{3:} \oplus \left[\mathbf{F}_{\underline{k}}^{(0)}\right]_{31} \left[\mathbf{A}_{\underline{k}}^{(0)}\right]_{1:} \\ &= [\varepsilon \quad \varepsilon \quad d_3(k)] \\ &\quad \oplus d_1(k+1)[d_1(k) \quad \varepsilon \quad \varepsilon] \\ &= [d_1(k)d_1(k+1) \quad \varepsilon \quad d_3(k)]\end{aligned}$$

$$\left[\mathbf{F}_{\underline{k}}^{(1)}\right]_{3:} = [\varepsilon \quad d_2(k+1) \quad \varepsilon]$$

$$\begin{aligned}\left[\mathbf{B}_{\underline{k}}^{(1)}\right]_{3:} &= \left[\mathbf{B}_{\underline{k}}^{(0)}\right]_{3:} \oplus \left[\mathbf{F}_{\underline{k}}^{(0)}\right]_{31} \left[\mathbf{B}_{\underline{k}}^{(0)}\right]_{1} \\ &= [\varepsilon \quad \varepsilon] \oplus d_1(k+1)[e \quad \varepsilon] \\ &= [d_1(k+1) \quad \varepsilon]\end{aligned}$$

Subsequently, substituting $x_2(k+1)$ for $x_3(k+1)$ results in

$$\begin{aligned}\left[\mathbf{A}_{\underline{k}}^{(2)}\right]_{3:} &= \left[\mathbf{A}_{\underline{k}}^{(1)}\right]_{3:} \oplus \left[\mathbf{F}_{\underline{k}}^{(1)}\right]_{32} \left[\mathbf{A}_{\underline{k}}^{(1)}\right]_{2:} \\ &= [d_1(k)d_1(k+1) \quad \varepsilon \quad d_3(k)] \\ &\quad \oplus d_2(k+1)[\varepsilon \quad d_2(k) \quad \varepsilon] \\ &= [d_1(k)d_1(k+1) \quad d_2(k)d_2(k+1) \quad d_3(k)]\end{aligned}$$

$$\left[\mathbf{F}_{\underline{k}}^{(2)}\right]_{3:} = [\varepsilon \quad \varepsilon \quad \varepsilon]$$

$$\begin{aligned}\left[\mathbf{B}_{\underline{k}}^{(2)}\right]_{3:} &= \left[\mathbf{B}_{\underline{k}}^{(1)}\right]_{3:} \oplus \left[\mathbf{F}_{\underline{k}}^{(1)}\right]_{32} \left[\mathbf{B}_{\underline{k}}^{(1)}\right]_{2} \\ &= [d_1(k+1) \quad \varepsilon] \oplus d_2(k+1)[\varepsilon \quad e] \\ &= [d_1(k+1) \quad d_2(k+1)]\end{aligned}$$

Consequently, $x_3(k+1)$ turns into an eliminated variable, and hence all the state variables are eliminated. Meanwhile, since $x_1(k+1)$ and $x_2(k+1)$ are invariant through these substitutions, $[\boldsymbol{A}_{\underline{k}}^{(w)}]_{i:}$, $[\boldsymbol{F}_{\underline{k}}^{(w)}]_{i:}$, and $[\boldsymbol{B}_{\underline{k}}^{(w)}]_{i:}\quad (0 \le w \le 2,\ i \neq 3)$ are also invariant. Utilizing Eq. (56), they are represented in the following way:

$$\left.\begin{aligned}[\boldsymbol{A}_{\underline{k}}^{(w)}]_{1:} &= [d_1(k) \quad \varepsilon \quad \varepsilon] \\ [\boldsymbol{A}_{\underline{k}}^{(w)}]_{2:} &= [\varepsilon \quad d_2(k) \quad \varepsilon]\end{aligned}\right. \ ,\quad (0 \le w \le 2)$$

$$[\boldsymbol{F}_{\underline{k}}^{(w)}]_{i:} = [\varepsilon \quad \varepsilon \quad \varepsilon]\ ,\quad (i = 1,2,\ 0 \le w \le 2)$$

$$\left.\begin{aligned}[\boldsymbol{B}_{\underline{k}}^{(w)}]_{1:} &= [e \quad \varepsilon] \\ [\boldsymbol{B}_{\underline{k}}^{(w)}]_{2:} &= [\varepsilon \quad e]\end{aligned}\right. \ ,\quad (0 \le w \le 2)$$

It follows that $[\boldsymbol{A}_{\underline{k}}^{(2)}]$ and $[\boldsymbol{B}_{\underline{k}}^{(2)}]$ coincide with $\boldsymbol{A}$ in Eq. (17) and $\boldsymbol{B}$ in Eq. (18), respectively. Hence, the state-space equation in the form of Eq. (43) is transformed into Eq. (23).

Considering Eqs. (45)) ~ (48) and Eqs. (53)) ~ (56), each element of the system matrices is expressed as a polynomial of the system parameters. Therefore, the internal representation of the system matrices should be designed in order to store the degrees of the system parameters, which is defined in the next subsection.

4.3 Internal Representation of System Matrices

In this subsection, the internal representation of the system matrices is considered.

As multiplication or exponentials of the system parameters $d_i(k+j)\ (1 \le i \le n, 0 \le j \le 1)$ are incorporated into $\boldsymbol{A}_{\underline{k}}^{(w)}$, $\boldsymbol{F}_{\underline{k}}^{(w)}$, $\boldsymbol{B}_{\underline{k}}^{(w)}$ and $\boldsymbol{C}_{\underline{k}}$ in their elements, the internal representation of the system matrices should be designed such that exponentials of the variables are effectively expressed.

Therefore, the entity shown in Figure 2 is proposed. It is referred to as a ‘plane’ hereafter. In Figure 2, $S_{ij} \in \boldsymbol{Z} \quad (1 \le i \le n, 0 \le j \le N)$ stores the degree of the corresponding system parameter $d_i(k+j)$, where $\boldsymbol{Z}$ represents the non-negative integer field. Recall that N is a prediction step number, and n denotes the number of system parameters. Using this plane, a variable space defined in

$$\bigotimes_{j=0}^{N} \bigotimes_{i=1}^{n} d_i(k+j)^{S_{ij}} \tag{57}$$

can be expressed, and let this field be denoted as $\boldsymbol{S}$. An example of $d_1(k) \otimes d_2(k+1)^2$ is illustrated in Figure 3.

Multiplication of the system parameters is expressed using a single plane, and parameters combined with $\oplus$ are expressed by a collection of multiple planes. If the number of planes is zero, this represents ε. If the number of planes is one and all the elements of the plane are zero, this represents e.

Concerning the properties of the max-plus algebra, basic operations for the system parameters can be performed by utilizing the plane. The operation rules are explained in Table 2.

Utilizing the rules shown in Table 2, each element of the system matrices is expressed by a collection of planes. In order to confirm this fact, the following lemma is considered.

Lemma 1.

When $\boldsymbol{v}$ and $\boldsymbol{w}$ are expressed by a collection of planes, $\boldsymbol{vw}$ is also expressed by a collection of planes.

S_{ij}

$d_1(k)$	$d_1(k+1)$	$d_1(k+2)$	$\cdots$	$d_1(k+N)$
$d_2(k)$	$d_2(k+1)$	$d_2(k+2)$		$d_2(k+N)$
$\vdots$			$\ddots$	$\vdots$
$d_n(k)$	$d_n(k+1)$	$d_n(k+2)$	$\cdots$	$d_n(k+N)$

Figure 2. Plane for expressing system parameters ($\boldsymbol{S}$)

S_{ij}

1	0	0	$\cdots$	0
0	2	0		0
$\vdots$			$\ddots$	$\vdots$
0	0	0	$\cdots$	0

Figure 3. Representation of $d_1(k) \otimes d_2(k+1)^2$

Table 2. Operation rules for plane

1	Addition of two planes A collection of source planes is equivalent to a result. If the sources are equal, they are reduced to a single plane.
2	Multiplication of two planes Summations of respective elements are a result.
3	Increment of k Shift $\boldsymbol{S}_{ij}$ a single column to the right, and fill the first column with 0.

Proof.

$\boldsymbol{v}$ and $\boldsymbol{w}$ are represented as

$$\boldsymbol{v} = \bigoplus_{i=1}^{I} \boldsymbol{v}_i \ , \quad \boldsymbol{w} = \bigoplus_{j=1}^{J} \boldsymbol{w}_j \qquad \boldsymbol{v}_i \in \boldsymbol{S},\ \boldsymbol{w}_j \in \boldsymbol{S} \tag{58}$$

Utilizing the distributive law,

$$\boldsymbol{vw} = \bigoplus_{i=1}^{I} \bigoplus_{j=1}^{J} \left(\boldsymbol{v}_i \boldsymbol{w}_j\right) = \bigoplus_{l=1}^{L} \left(\boldsymbol{z}_l\right) \qquad (L = I \cdot J) \tag{59}$$

Using the multiplication rule 2 in Table 2, $\boldsymbol{z}_l = \boldsymbol{v}_i \boldsymbol{w}_j$ is expressed by a single plane. Thus, $\boldsymbol{vw}$ is expressed by a collection of planes.

Since each element of the system matrices in Eqs. (45)) ~ (48) is represented by a power of the system parameters, all the elements of the system matrices in Eqs. (43)) ~ (44) are expressed by a collection of planes. Moreover, concerning the recursive equations Eqs. (53)) ~ (55) and utilizing Lemma 1, it follows that all the elements of the system matrices in Eqs. (23)) ~ (24) can be expressed as a collection of the system parameters.

$$[\boldsymbol{A}_{\underline{k}}^{(1)}]_{3:} = [\varepsilon \quad \varepsilon \quad d_3(k)] \oplus d_1(k+1) \otimes [d_1(k) \quad \varepsilon \quad \varepsilon]$$

$$= \left[\{\} \quad \{\} \quad \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline 1 & 0 \\ \hline\end{array} \right] \oplus \begin{array}{|c|c|}\hline 0 & 1 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline\end{array} \otimes \left[\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline\end{array} \quad \{\} \quad \{\} \right]$$

$$= \left[\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline\end{array} \quad \{\} \quad \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline 1 & 0 \\ \hline\end{array} \right]$$

$$= [d_1(k)d_1(k+1) \quad \varepsilon \quad d_3(k)]$$

$$[\boldsymbol{A}_{\underline{k}}^{(2)}]_{3:}$$

$$= [d_1(k)d_1(k+1) \quad \varepsilon \quad d_3(k)] \oplus d_2(k+1) \otimes [\varepsilon \quad d_2(k) \quad \varepsilon]$$

$$= \left[\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline\end{array} \quad \{\} \quad \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline 1 & 0 \\ \hline\end{array} \right] \oplus \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 1 \\ \hline 0 & 0 \\ \hline\end{array} \otimes \left[\{\} \quad \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 1 & 0 \\ \hline 0 & 0 \\ \hline\end{array} \quad \{\} \right]$$

$$= \left[\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline\end{array} \quad \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 1 & 1 \\ \hline 0 & 0 \\ \hline\end{array} \quad \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline 1 & 0 \\ \hline\end{array} \right]$$

$$= [d_1(k)d_1(k+1) \quad d_2(k)d_2(k+1) \quad d_3(k)]$$

Figure 4. Calculations of $[\boldsymbol{A}_{\underline{k}}^{(1)}]_{3:}$ and $[\boldsymbol{A}_{\underline{k}}^{(2)}]_{3:}$

Therefore, the state-space equations Eqs. (23)) ~ (24) are derived and the elements of the system matrices are expressed by a collection of planes.

As an example, calculations of $[\boldsymbol{A}_{\underline{k}}^{(1)}]_{3:}$ and $[\boldsymbol{A}_{\underline{k}}^{(2)}]_{3:}$ using planes are shown in Figure 4. $\{\}$ represents ε, and indicates that the number of planes is zero.

4.4 Internal representation of optimal control input

This subsection considers an internal representation of the optimal control input shown in Eqs. (35)) ~ (36).

Since $\boldsymbol{A}_{\underline{k}}$, $\boldsymbol{B}_{\underline{k}}$ and $\boldsymbol{C}_{\underline{k}}$ are expressed by a collection of planes, $\boldsymbol{A}_{\underline{k+j-1}}$, $\boldsymbol{B}_{\underline{k+j-1}}$ and $\boldsymbol{C}_{\underline{k+j}}$ $(1 \le j \le N)$ are also represented by utilizing the operation rule 3) in Table 2. In addition, Eq. (9) indicates that when all elements of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are expressed by a collection of planes, all elements of $\boldsymbol{AB}$ are also expressed by a collection of planes. Thus, all the elements of $\boldsymbol{\Gamma}_{\underline{k}}$ and $\boldsymbol{\Delta}_{\underline{k}}$ in Eqs. (28)) ~ (29) are expressed by a collection of planes.

In this way, the internal representation of the optimal control input in Eqs. (35)~(36) is derived. It follows that the inputs are expressed as functions of the system parameters.

$$\boldsymbol{\Gamma}_{\underline{k}} = \boldsymbol{C}_{\underline{k+1}} \boldsymbol{A}_{\underline{k}}$$

$$= \begin{bmatrix} \{\} & \{\} & \begin{array}{|c|c|}\hline 0&0\\\hline 0&0\\\hline 0&1\\\hline\end{array} \end{bmatrix} \otimes \begin{bmatrix} \begin{array}{|c|c|}\hline 1&0\\\hline 0&0\\\hline 0&0\\\hline\end{array} & \{\} & \{\} \\ \{\} & \begin{array}{|c|c|}\hline 0&0\\\hline 0&1\\\hline 0&0\\\hline\end{array} & \{\} \\ \begin{array}{|c|c|}\hline 1&1\\\hline 0&0\\\hline 0&0\\\hline\end{array} & \begin{array}{|c|c|}\hline 0&0\\\hline 1&1\\\hline 0&0\\\hline\end{array} & \begin{array}{|c|c|}\hline 0&0\\\hline 0&0\\\hline 1&0\\\hline\end{array} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{array}{|c|c|}\hline 1&1\\\hline 0&0\\\hline 0&1\\\hline\end{array} & \begin{array}{|c|c|}\hline 0&0\\\hline 1&1\\\hline 0&1\\\hline\end{array} & \begin{array}{|c|c|}\hline 0&0\\\hline 0&0\\\hline 1&1\\\hline\end{array} \end{bmatrix}$$

$$= \begin{bmatrix} d_1(k)d_1(k+1)d_3(k+1) \\ d_2(k)d_2(k+1)d_3(k+1) \\ d_3(k)d_3(k+1) \end{bmatrix}^T$$

$$\boldsymbol{\Delta}_{\underline{k}} = \boldsymbol{C}_{\underline{k+1}} \boldsymbol{B}_{\underline{k}}$$

$$= \begin{bmatrix} \{\} & \{\} & \begin{array}{|c|c|}\hline 0&0\\\hline 0&0\\\hline 0&1\\\hline\end{array} \end{bmatrix} \otimes \begin{bmatrix} \begin{array}{|c|c|}\hline 0&0\\\hline 0&0\\\hline 0&0\\\hline\end{array} & \{\} \\ \{\} & \begin{array}{|c|c|}\hline 0&0\\\hline 0&0\\\hline 0&0\\\hline\end{array} \\ \begin{array}{|c|c|}\hline 0&1\\\hline 0&0\\\hline 0&0\\\hline\end{array} & \begin{array}{|c|c|}\hline 0&0\\\hline 0&1\\\hline 0&0\\\hline\end{array} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{array}{|c|c|}\hline 0&1\\\hline 0&0\\\hline 0&1\\\hline\end{array} & \begin{array}{|c|c|}\hline 0&0\\\hline 0&1\\\hline 0&1\\\hline\end{array} \end{bmatrix}$$

$$= \begin{bmatrix} d_1(k+1)d_3(k+1) \\ d_2(k+1)d_3(k+1) \end{bmatrix}^T$$

Figure 5. Calculation of $\boldsymbol{\Gamma}_{\underline{k}}$ and $\boldsymbol{\Delta}_{\underline{k}}$ for $N = 1$

An illustrative calculation of $\boldsymbol{\Gamma}_{\underline{k}}$ and $\boldsymbol{\Delta}_{\underline{k}}$ for $N = 1$ using planes is shown in Figure 5. The correctness of this result is confirmed in the next section.

In on-line calculation, the inputs are determined by substituting the current values of the system parameters, state variables, and the reference inputs.

5. EXECUTION EXAMPLES

In order to confirm the effectiveness of this proposed method, this section presents execution examples. Note that the experiments are performed on a Hitachi Pentium-4 2.4GHz PC.

5.1 demonstrates an internal representation of the production system, and 5.2 estimates the calculation volume for the derivation of the state-space equation.

5.1 Results of Internal Representation

Figure 6 shows the result of deriving a state-space equation for the system depicted in Figure 1. Numbers inside parentheses denote step numbers appended to the event counter *k*. For instance, d2(1) represents $d_2(k+1)$.

Figure 7 shows the representations of $\boldsymbol{\Gamma}_{\underline{k}}$ and $\boldsymbol{\Delta}_{\underline{k}}$ for the prediction step N =1, and indicates

$$\boldsymbol{\Gamma}_{\underline{k}} = \begin{bmatrix} d_1(k)d_1(k+1)d_3(k+1) \\ d_2(k)d_2(k+1)d_3(k+1) \\ d_3(k)d_3(k+1) \end{bmatrix}^T \tag{60}$$

$$\boldsymbol{\Delta}_{\underline{k}} = \begin{bmatrix} d_1(k+1)d_3(k+1) & d_2(k+1)d_3(k+1) \end{bmatrix} \tag{61}$$

Figure 8 shows the representations of $\boldsymbol{\Delta}_{\underline{k}}$ for N =3. For instance, the third element indicates

$$\begin{aligned} [\boldsymbol{\Delta}_{\underline{k}}]_{31} &= d_1(k+1)d_1(k+2)d_1(k+3)d_3(k+3) \\ &\oplus d_1(k+1)d_1(k+2)d_3(k+2)d_3(k+3) \\ &\oplus d_1(k+1)d_3(k+1)d_3(k+2)d_3(k+3) \end{aligned} \tag{62}$$

5.2 Calculation Volume

As is shown in the previous subsection, the representation of system matrices becomes more complicated in proportion to the value of the prediction step number *N*. Since N should be set to be a large number for designing a robust controller, this algorithm is effective in such situations.

To confirm this, the total number of planes for $\boldsymbol{\Gamma}_{\underline{k}}$ and $\boldsymbol{\Delta}_{\underline{k}}$ are examined and shown in Figure 9. The total number of planes is defined as a summation of the number of planes in each element. In Figure 9, it seems that the number of planes grows quadratically as the prediction step N increases. In fact, the growth is estimated to be n^2+2n for $\underline{\boldsymbol{\Gamma}}_k$, and n^2+n for $\underline{\boldsymbol{\Delta}}_k$, respectively.

Figure 10 shows the computation times for deriving the internal representation of the state-space equation and the optimal control input. The former is the time for obtaining the internal representation of $\underline{\boldsymbol{A}}_k$, $\underline{\boldsymbol{B}}_k$ and $\underline{\boldsymbol{C}}_k$ in Eqs. (23)) ~ (24), and the latter is for $\underline{\boldsymbol{\Gamma}}_k$ and $\underline{\boldsymbol{\Delta}}_k$ in Eq. (35). Each result represents the time taken averaged over ten executions. As for the derivation of the state-space equation, the calculation volume is fixed and hence the computation times are almost the same. On the other hand, the computation times for deriving the optimal control input grow at a greater than quadratic rate as the prediction step number increases.

```
A =
    'd1(0)'                   ''              ''
                 ''       'd2(0)'             ''
    'd1(0)d1(1)'    'd2(0)d2(1)'    'd3(0)'
B =
    'e'              ''
         ''      'e'
    'd1(1)'    'd2(1)'
C =
     ''      ''      'd3(0)'
```

Figure 6. Representations of $\underline{\boldsymbol{A}}_k$, $\underline{\boldsymbol{B}}_k$ and $\underline{\boldsymbol{C}}_k$

```
Gamma =
  'd1(0)d1(1)d3(1)' 'd2(0)d2(1)d3(1)'   'd3(0)d3(1)'
Delta =
  'd1(1)d3(1)' 'd2(1)d3(1)'
```

Figure 7. Representations of $\underline{\boldsymbol{\Gamma}}_k$ and $\underline{\boldsymbol{\Delta}}_k$ (for $N=1$)

```
Delta(1, 1) =
    'd1(1)d3(1)'
Delta(2, 1) =
    'd1(1)d1(2)d3(2)(+)d1(1)d3(1)d3(2)'
Delta(3, 1) =
   'd1(1)d1(2)d1(3)d3(3)(+)d1(1)d1(2)d3(2)d3(3)
     (+)d1(1)d3(1)d3(2)d3(3)'
Delta(1, 2) =
    'd2(1)d3(1)'
Delta(2, 2) =
    'd2(1)d2(2)d3(2)(+)d2(1)d3(1)d3(2)'
Delta(3, 2) =
   'd2(1)d2(2)d2(3)d3(3)(+)d2(1)d2(2)d3(2)d3(3)
     (+)d2(1)d3(1)d3(2)d3(3)'
```

Figure 8. Representations of $\underline{\boldsymbol{\Delta}}_k$ (for $N=3$)

Therefore, it follows that deriving an output prediction equation manually is impractical when N is large, which highlights the usefulness of this proposed algorithm.

6. CONCLUDING REMARKS

This paper proposes a new algorithm for deriving a state-space equation and determining an optimal control input in MPL systems.

Utilizing the proposed method, the derivation processes can be achieved by providing only constraint matrices and parameter vectors, while they are handled manually in current research. Moreover, since the system parameters are handled as variables in the solution of the optimal control input, the input can be calculated by substituting the internal states and the reference signals.

Through execution examples, internal representations of

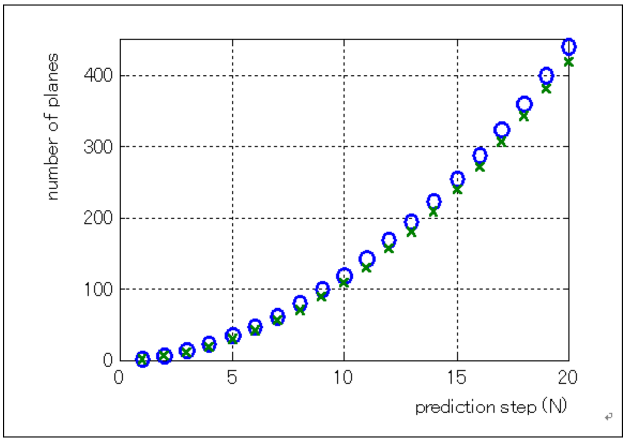

Figure 9. Number of planes for $\underline{\boldsymbol{\Gamma}}_k$ (o) and $\underline{\boldsymbol{\Delta}}_k$ (x)

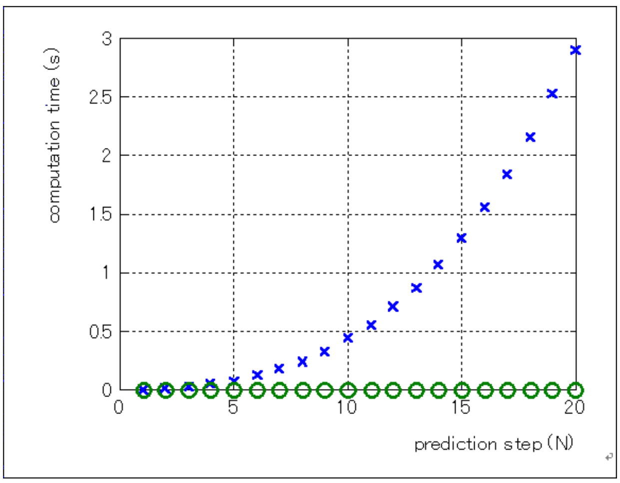

Figure 10. Computation times for deriving the state-space equation (o) and the optimal control input (x)

the system constraints and the output prediction equations are shown, and the effectiveness of this proposed algorithm is confirmed by examining the calculation volume for a controller based on MPC.

As this algorithm is applicable to complicated systems, analyses and control designs for practical systems using the max-plus algebra are becoming realistic.

REFERENCES

Baccelli, F., Cohen, G., Olsder, G., and Quadrat, J. (1992) *Synchronization and Linearity*, New York, Jhon Wiley & Son.

Boimond, J. and Ferrier, J. (1996) Internal model control and max-algebra: Controller design, *IEEE Transactions on Automatic Control*, ***41***, 457-461.

Cohen, G., Moller, P., Quadrat, J. , and Viot, M. (1989) Algebraic tools for the performance evaluation of discrete event systems, *Proceedings of the IEEE* ***77,*** 39-58.

Goto, H., Masuda, S., Takeyasu, K., Amemiya, T. (2002) Model Predictive Control for Production Systems Based On Max-Plus Algebra, *Industrial Engineering and Management Systems*, ***1***, 1-9.

Masuda, S., Goto, H., Amemiya, T., and Takeyasu, K. (2003) An optimal inverse system for max-plus linear systems with linear parameter-varying structure (In Japanese), *Transactions of ISCIE*, ***16***, 44-53.

Schullerus, G. and Krebs, V. (2001) Diagnosis of batch processes based on parameter estimation of discrete event models, *Proceedings of the European Control Conference*, 1612-1617.

Schutter, B. and Boom, T. (2001) Model predictive control for max-plus-linear systems, *Automatica*, ***37***, 1049-1056.