IEMS Vol. 7, No. 1, pp. 23-33, June 2008.

Monitoring and Scheduling Methods for MIMO-FIFO Systems Utilizing Max-Plus Linear Representation

Hiroyuki Goto†
Department of Management Information Systems Science
Nagaoka University of Technology Nagaoka, Niigata, 940-2188, JAPAN
Tel: +81-258-47-9322, Email: hgoto@kjs.nagaokaut.ac.jp

Shiro Masuda
Faculty of System Design
Tokyo Metropolitan University, Hino, Tokyo, 161-0065, JAPAN
Tel: +81-42-585-8631, Email smasuda@cc.tmit.ac.jp

Received Date, August 2006; Accepted Date, March 2008

Abstract. This paper proposes an approach to monitoring and scheduling methods for repetitive MIMO-FIFO DESs. We use max-plus algebra for modeling and formulation, known as an effective approach for controller design for this type of system. Because a certain type of linear equations in max-plus algebra can represent the system's behavior, the principal concerns in past researches were how to solve the equations. However, the researches focused mainly on analyses of the relation between inputs and outputs of the system, which implies that the changes or the slacks of internal states were not clarified well. We first examine several properties of the corresponding state variables, which contribute to finding and tracing the float times in each process. Moreover, we provide a rescheduling method that can take into account delays or changes of the internal states. These methods would be useful in schedule control or progress management.

Keywords: Max-Plus Linear System, Earliest/Latest Starting Time, Bottleneck, Rescheduling, MIMO, FIFO.

1. INTRODUCTION

This paper proposes monitoring and scheduling methods for DESs (Discrete Event Systems) with repetitive MIMO (Multiple Inputs and Multiple Outputs)-FIFO (First In, First Out) structure. A specific algebra called max-plus algebra (Cohen *et al.*, 1989; Baccelli *et al.*, 1992) performs the fundamental formulation. A class of linear equations in max-plus algebra, called the MPL (Max-Plus Linear) form, represents the behavior of this sort of system.

In schedule control for production systems or progress management for projects, a variety of time constraints is imposed in many cases. Relevant processes cannot start manufacturing until a previous process has completed processing, multiple processes can manufacture concurrently, etc. In these kinds of management, it is necessary to take care where there are bottlenecks among internal processes, and sufficient attention paid to progress so they are not late for due dates. Approaches based on TPN (Timed Petri Net) (Ramamoorthy and Ho, 1980) are proposed for describing and analyzing systems where available resources are repeatedly used. They treat the firing times of transitions as the starting or completion times of processes, and the holding times of tokens as the execution times in the respective processes. Control places represent no-concurrency of internal resources. A subclass of TPN called TEG (Timed Event Graph) (Cohen *et al.*, 1989) can describe the behavior of repetitive MIMO-FIFO systems, and it can also be formulated in MPL form.

Max-plus algebra, also known as $(\max, +)$ algebra, has addition equivalent to the max operation and multiplication to plus operation. It follows the effective properties in conventional $(+, \times)$ algebra such as the commutative and distributive laws. Since the formulation in MPL form is similar to the state space representation in modern control theory in conventional algebra, previous works in control theory have been applied to MPL systems; internal model control (Boimond and Ferrier, 1996), adaptive control (Menguy *et al.*, 2000), model predictive control (Schutter and Boom, 2001), etc. Goto and Masuda (2004a)

† : Corresponding Author

proposed an algorithm for deriving the state space representation in which the system parameters are handled as variables. A scheduling problem regarding feeding times for a class of MIMO-FIFO systems has been resolved by this research.

However, the relevant researches focus on determining the desirable input times, and no method has been proposed for supervising midstream processes. By keeping track of internal states of the system, it is expected that bottleneck processes or processes that need attention can be found easily. Therefore, this paper proposes general and efficient methods for identifying and predicting the internal states, focusing on repetitive MIMO-FIFO systems. The essential contents of this paper include:

- Deriving the state space representation in MPL form by giving execution times in respective processes and precedence constraints among internal processes.
- Utilizing the state space equation and the output equation, calculate the earliest/latest starting times and the total float in each process, and find bottleneck processes.
- Rescheduling method when the relevant parameters such as the feeding times or processing times have changed after the processing has started.

Another existing method based on PERT (Program Evaluation and Review Technique) or CPM (Critical Path Method) (Hillier and Lieberman, 2002) theories is well known and focuses on calculating total float or finding bottlenecks. However, this method is designed for treating SISO (Single Input and Single Output) systems, which means that it cannot always gives an effective solution for MIMO structure and/or the processing of the next batch that can start before the current batch has finished. In such cases, the bottlenecks are possibly in different locations. On the other hand, MPL systems can handle MIMO systems and describe the no-concurrency in internal resources. Thus, the methods proposed in this paper are also extensions of PERT whose application scope is broadened.

2. MAX-PLUS LINEAR SYSTEM

This section reviews max-plus algebra and the MPL systems which are the essential backgrounds throughout this research.

2.1 Max-Plus Algebra

Max-plus algebra is an algebraic system that is suitable for describing a certain class of discrete event systems. In $\mathcal{D} = \boldsymbol{R} \cup \{-\infty\}$, operators for addition and multiplication are defined as:

$$x \oplus y = \max(x,\ y)\,,\quad x \otimes y = x + y\,,$$

where $\boldsymbol{R}$ is the real field. The $\otimes$ operator is often suppressed when no confusion is likely to arise. These hold the commutative, associative and distributive laws. By defining unit elements for these operators as $\varepsilon(=-\infty)$ and $e(=0)$ respectively, the following relations are satisfied for an arbitrary $x \in \mathcal{D}$:

$$x \oplus \varepsilon = \varepsilon \oplus x = x\,,\quad x \otimes e = e \otimes x = x\,.$$

In addition, the following two operators are defined for subsequent discussions:

$$x \wedge y = \min(x,\ y)\,,\quad x \backslash y = -x + y\,.$$

Operators for multiple numbers are; if $m \le n$, then

$$\bigoplus_{k=m}^{n} x_k = x_m \oplus x_{m+1} \oplus \cdots \oplus x_n = \max(x_m,\ x_{m+1}, \cdots,\ x_n)$$

$$\bigwedge_{k=m}^{n} x_k = x_m \wedge x_{m+1} \wedge \cdots \wedge x_n = \min(x_m,\ x_{m+1}, \cdots,\ x_n)$$

Definitions for matrices are: for $\boldsymbol{X} \in \mathcal{D}^{m\times n}$, $\boldsymbol{X}^T$ is the transpose matrix of $\boldsymbol{X}$, and $[\boldsymbol{X}]_{ij}$ represents its $(i,\ j)$ th element. For $\boldsymbol{X},\ \boldsymbol{Y} \in \mathcal{D}^{m\times n}$,

$$[\boldsymbol{X} \oplus \boldsymbol{Y}]_{ij} = [\boldsymbol{X}]_{ij} \oplus [\boldsymbol{Y}]_{ij} = \max([\boldsymbol{X}]_{ij}, [\boldsymbol{Y}]_{ij})$$

$$[\boldsymbol{X} \wedge \boldsymbol{Y}]_{ij} = [\boldsymbol{X}]_{ij} \wedge [\boldsymbol{Y}]_{ij} = \min([\boldsymbol{X}]_{ij}, [\boldsymbol{Y}]_{ij})$$

If $\boldsymbol{X} \in \mathcal{D}^{m\times l}$, $\boldsymbol{Y} \in \mathcal{D}^{l\times p}$,

$$[\boldsymbol{X} \otimes \boldsymbol{Y}]_{ij} = \bigoplus_{k=1}^{l} \left([\boldsymbol{X}]_{ik} \otimes [\boldsymbol{Y}]_{kj}\right) = \max_{k=1,\cdots,l}([\boldsymbol{X}]_{ik} + [\boldsymbol{Y}]_{kj}),$$

$$[\boldsymbol{X} \odot \boldsymbol{Y}]_{ij} = \bigwedge_{k=1}^{l} \left([\boldsymbol{X}]_{ik} \backslash [\boldsymbol{Y}]_{kj}\right) = \min_{k=1,\cdots,l}(-[\boldsymbol{X}]_{ik} + [\boldsymbol{Y}]_{kj}).$$

Unit elements for matrices are: $\boldsymbol{\varepsilon}_{mn}$ is a matrix whose all elements are ε in $\boldsymbol{\varepsilon}_{mn} \in \mathcal{D}^{m\times n}$, and $\boldsymbol{e}_m$ is a matrix whose diagonal elements are e and all off-diagonal elements are ε in $\boldsymbol{e}_m \in \mathcal{D}^{m\times m}$. In $\boldsymbol{x},\ \boldsymbol{y} \in \mathcal{D}^m$, if $[\boldsymbol{x}]_i \le [\boldsymbol{y}]_i$ holds for all i $(1 \le i \le m)$, we write $\boldsymbol{x} \le \boldsymbol{y}$ simply.

2.2 MPL System

The MPL system is defined as a system whose behavior can be described in linear form in max-plus algebra. It is similar to the state-space equations in modern control theory, stated as follows:

$$\boldsymbol{x}(k) = \boldsymbol{A}\boldsymbol{x}(k-1) \oplus \boldsymbol{B}\boldsymbol{u}(k), \tag{1}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k). \tag{2}$$

where k is called the event counter that represents the number of event occurrences from the initial state. Recall here that the $\otimes$ operators are omitted for simplicity; that is, $\boldsymbol{A}\boldsymbol{x}(k-1)=\boldsymbol{A}\otimes\boldsymbol{x}(k-1)$, etc. are followed. $\boldsymbol{x}(k)\in\mathcal{D}^n$, $\boldsymbol{u}(k)\in\mathcal{D}^p$, and $\boldsymbol{y}(k)\in\mathcal{D}^q$ represent the state variables, input variables, and output variables, respectively. n, p, and q are the corresponding dimensions. $A\in\mathcal{D}^{n\times n}$, $\boldsymbol{B}\in\mathcal{D}^{n\times p}$, and $\boldsymbol{C}\in\mathcal{D}^{q\times n}$ are called the system matrix, input matrix, output matrix, respectively.

An example to illustrate the kind of systems that can be formulated in MPL form utilizing equations (1) and (2). Figure 1 shows the machining sequence in a simple production system. Process No. 1 receives the raw material from the input lane, and then processes No.2 and No.3 manufacture the parts concurrently. Process No.4 receives the output from No.2 and No.3, fabricates them, and then sends the resulting part to the output lane. The processing times in sequences 1-4 are denoted as d_1, d_2, d_3, d_4, respectively. Now suppose each process has the following constraints:

- While the machines are at work, they cannot start processing for the subsequent parts.
- The processes No. 2-4, which have precedence constraints, cannot start processing until they have received the manufactured parts from the preceding processes.
- The process No. 1, which has an external input, cannot start processing until it receives the material.
- When the machine is empty, it starts processing as soon as all the required materials from the preceding processes and external inputs become available.

For the k th batch, suppose $u(k)$, $\boldsymbol{x}(k)$ and $y(k)$ represent the feeding times, processing start times, and finishing times, respectively. It follows that the following relations hold:

$$x_1(k)=\max\{u(k), x_1(k-1)+d_1\} \tag{3}$$

$$x_2(k)=\max\{x_2(k-1)+d_2, x_1(k)+d_1\} \tag{4}$$

$$x_3(k)=\max\{x_3(k-1)+d_3, x_1(k)+d_1\} \tag{5}$$

$$x_4(k)=\max\{x_4(k-1)+d_4, x_2(k)+d_2,\ x_3(k)+d_3) \tag{6}$$

$$y(k)=x_4(k)+d_4 \tag{7}$$

By substituting equation (3) into equations (4) and (5), and these into equation (6), equations (3)-(7) can be described as forms of equations (1) and (2), where

$$\boldsymbol{A}=\begin{pmatrix} d_1 & \varepsilon & \varepsilon & \varepsilon \\ d_1^2 & d_2 & \varepsilon & \varepsilon \\ d_1^2 & \varepsilon & d_3 & \varepsilon \\ a_{41} & d_2^2 & d_3^2 & d_4 \end{pmatrix},\quad \boldsymbol{B}=\begin{pmatrix} e \\ d_1 \\ d_1 \\ b_4 \end{pmatrix},,\quad \boldsymbol{C}=\begin{pmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \\ d_4 \end{pmatrix}^T, \tag{8}$$

$$a_{41}=d_1^2(d_2\oplus d_3),\quad b_4=d_1(d_2\oplus d_3). \tag{9}$$

This derivation uses the model of a production system shown here, but it can also be applied to project scheduling problems that have precedence constraints. In this case, the processes correspond to tasks. This paper hereafter uses the term of 'process' unless otherwise noted.

Since there is a term $x_i(k)$ on the right hand side of equations (4)-(6), they must be transformed into equations without the term $\boldsymbol{x}(k)$. This implies that they are expressed in the form of equation (1) which has been performed manually in previous works. Thus, there have been few discussions about the domain where equations describe and the general form of the system matrix and input/output matrices. Therefore, the next section makes a general inspection of the MPL equations for systems with precedence constraints or synchronizations.

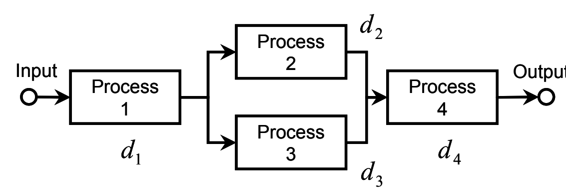

Figure 1. A simple manufacturing process

3. DERIVATION OF MPL REPRESENTATION

Now we inspect a process for deriving the MPL representation for a certain class of discrete event systems. First, we assume the relevant constraints are imposed on focused system in the following way.

- The number of processes is n, the number for external inputs is p, and q is for the number of external outputs.
- All machines are used only once for a single batch.
- The subsequent batch cannot start processing when the machine is at work with the current one.
- Processes that have precedence constraints cannot start processing until they have received the required parts from preceding processes.
- For processes that have external inputs, processing cannot start until all required materials have arrived.
- The processing starts as soon as all conditions above are satisfied.

Let $[\boldsymbol{x}(k)]_i$ and $d_i(k)\,(\geq 0)$ $(1\leq i\leq n)$ be the starting time and the processing time for each process, and the initial condition be $\boldsymbol{x}(0)=\boldsymbol{\varepsilon}_{n1}$. Matrices $\boldsymbol{A}_k^0$, $\boldsymbol{F}_k$, $\boldsymbol{B}_k^0$ and $\boldsymbol{C}_k$ are introduced for representing the structure of systems as follows:

$$[\boldsymbol{A}_k^0]_{ij}=\begin{cases} d_i(k) & : \text{If } i=j. \\ \varepsilon & : \text{Otherwise.} \end{cases}$$

$$[\boldsymbol{F}_k]_{ij}=\begin{cases} d_j(k) & : \text{If process } i \text{ has a preceding process } j. \\ \varepsilon & : \text{If process } i \text{ does not have a preceding process } j. \end{cases}$$

$$[\boldsymbol{B}^0]_{ij}=\begin{cases} e & : \text{If process } i \text{ has an external input } j. \\ \varepsilon & : \text{If process } i \text{ does not have an external input } j. \end{cases}$$

$$[\boldsymbol{C}_k]_{ij}=\begin{cases} d_j(k) & : \text{If process } j \text{ has an external output } i. \\ \varepsilon & : \text{If process } j \text{ does not have an external output } i. \end{cases}$$

$\boldsymbol{F}_k$ is referred to as the adjacency matrix. Hereafter, if process i has a preceding process j, we express this as $j \rightarrow i$. Let us inspect the above matrices and their products of relevant variables. Consider the following four elements:

$$\begin{aligned}[\boldsymbol{A}_{k-1}^0\boldsymbol{x}(k-1)]_i &= \oplus_{j=1}^n([\boldsymbol{A}_{k-1}^0]_{ij}\otimes[\boldsymbol{x}(k-1)]_j) \\ &= d_i(k-1)+[\boldsymbol{x}(k-1)]_i\end{aligned}, \tag{10}$$

$$\begin{aligned}[\boldsymbol{F}_k\boldsymbol{x}(k)]_i &= \oplus_{j=1}^n([\boldsymbol{F}_k]_{ij}\otimes[\boldsymbol{x}(k)]_j) \\ &= \begin{cases} \oplus_{j\in\mathcal{R}_i}(d_j(k)+[\boldsymbol{x}(k)]_j) & \text{if } \mathcal{R}_i\neq\{\phi\}, \\ \varepsilon & \text{if } \mathcal{R}_i=\{\phi\}\end{cases}\end{aligned} \tag{11}$$

$$\begin{aligned}[\boldsymbol{B}^0\boldsymbol{u}(k)]_i &= \oplus_{j=1}^p([\boldsymbol{B}^0]_{ij}\otimes[\boldsymbol{u}(k)]_j \\ &= \begin{cases} \oplus_{j\in\mathcal{P}_i}[\boldsymbol{u}(k)]_j & \textit{if } \mathcal{P}_i\neq\{\phi\}, \\ \varepsilon & \textit{if } \mathcal{P}_i=\{\phi\}\end{cases}\end{aligned} \tag{12}$$

$$\begin{aligned}[\boldsymbol{C}_k\boldsymbol{x}(k)]_i &= \oplus_{j=1}^n([\boldsymbol{C}_k]_{ij}\otimes[\boldsymbol{x}(k)]_j) \\ &= \begin{cases} \oplus_{j\in\mathcal{V}_i}(d_j(k)+[\boldsymbol{x}(k)]_j) & \text{if } \mathcal{V}_i\neq\{\phi\}, \\ \varepsilon & \text{if } \mathcal{V}_i=\{\phi\}\end{cases}\end{aligned} \tag{13}$$

where $\mathcal{R}_i$ and $\mathcal{P}_i$ represent the number set of preceding processes for process i and those for external inputs, respectively. In all systems that this paper handles, the following relation is satisfied:

$$\mathcal{R}_i=\{\phi\}\Rightarrow\mathcal{P}_i\neq\{\phi\}. \tag{14}$$

Note that there may possibly be cases where there are both preceding processes and external inputs, which implies that the inverse of equation (14) does not hold. In addition, $\mathcal{V}_i$ represents the number set of processes attached to output i. The right hand side of equation (10) is equal to the finishing time in each machine, and thus $\boldsymbol{A}_{k-1}^0\boldsymbol{x}(k-1)$ gives the finishing times for all machines. Concerning the right hand side of equation (11), it reveals that $\boldsymbol{F}_k\boldsymbol{x}(k)$ states the latest time among the finishing times in preceding processes. Regarding equation (12), $\boldsymbol{B}^0\boldsymbol{u}(k)$ is equal to the latest feeding time from external inputs. Moreover, considering equation (13), $\boldsymbol{C}_k\boldsymbol{x}(k)$ represents the latest time among the finishing times in processes attached to the corresponding output.

Utilizing these results, the starting time of the processing and the output time can be formulated as follows:

$$\boldsymbol{x}(k)=\boldsymbol{F}_k\boldsymbol{x}(k)\oplus\boldsymbol{A}_{k-1}^0\boldsymbol{x}(k-1)\oplus\boldsymbol{B}^0\boldsymbol{u}(k), \tag{15}$$

$$\boldsymbol{y}(k)=\boldsymbol{C}_k\boldsymbol{x}(k) \tag{16}$$

Note that equation (15) is similar to equation (1), however it includes $\boldsymbol{x}(k)$ in the right hand side. Thus, a process for transforming equation (15) into equation (1) is considered. For preparation, the following theorem regarding the adjacency matrix $\boldsymbol{F}_k$ is proved.

Theorem 1. There is an instance of $l\,(1\leq l\leq n)$ which satisfies $\boldsymbol{F}_k^{\,l}=\boldsymbol{\varepsilon}_{nn}$.

Proof. Assume the proposition does not hold, which indicates $\boldsymbol{F}_k^n\neq\boldsymbol{\varepsilon}_{nn}$. For simplicity, brief notations such as $[\boldsymbol{F}_k]_{ij}=f_{ij}$ and $[\boldsymbol{F}_k^l]_{ij}=f_{ij}^l$ are used. Utilizing the distributive law, the (i, j) th element of $\boldsymbol{F}_k^l$ can be expanded as:

$$\begin{aligned} f_{ij}^l &= \bigoplus_{k_1=1}^n f_{ik_1}^{l-1}\otimes f_{k_1j} \\ &= \bigoplus_{k_1=1}^n\left(\bigoplus_{k_2=1}^n f_{ik_2}^{l-2}\otimes f_{k_2k_1}\right)\otimes f_{k_1j}=\cdots \\ &= \bigoplus_{k_1=1}^n\bigoplus_{k_2=1}^n\cdots\bigoplus_{k_{l-1}=1}^n f_{ik_{l-1}}\otimes f_{k_{l-1}k_{l-2}} \\ &\quad \otimes\cdots\otimes f_{k_2k_1}\otimes f_{k_1j}\end{aligned} \tag{17}$$

Recalling that there are i and j that satisfy $f_{ij}^l\neq\varepsilon$, and utilizing the property of the operator $\oplus$, it leads to an existence of a set $\{f_{ik_{l-1}}, f_{k_{l-1}k_{l-2}}, \cdots, f_{k_2k_1}, f_{k_1j}\}$ which satisfies:

$$f_{ik_{l-1}}\neq\varepsilon,\; f_{k_{l-1}k_{l-2}}\neq\varepsilon,\cdots,f_{k_2k_1}\neq\varepsilon, f_{k_1j}\neq\varepsilon.$$

This means that there are the following precedence constraints:

$$j\rightarrow k_1, k_1\rightarrow k_2,\cdots,k_{l-2}\rightarrow k_{l-1}, k_{l-1}\rightarrow i.$$

Noting that the number of constraints is l and all machines are used only once, this means that there is at least one number set that satisfy:

$$1 \le \{i, j, k_1, k_2, \cdots, k_{l-1}\} \le n. \tag{18}$$

Since the number of elements in equation (18) is $l+1$, and all elements should be different each other, this kind of number set cannot be assigned if $l \ge n$. Therefore, this contradiction proves $\boldsymbol{F}_k^n = \boldsymbol{\varepsilon}_{nn}$. □

Now we go back to the transformation of equation (15). In equation (15), since $\boldsymbol{x}(k)$ appears in the first term on the right hand side, substituting equation (15) itself for the first term. Then, the following equation is obtained:

$$\begin{aligned}\boldsymbol{x}(k) &= \boldsymbol{F}_k[\boldsymbol{F}_k\boldsymbol{x}(k) \oplus \boldsymbol{A}_{k-1}^0\boldsymbol{x}(k-1) \oplus \boldsymbol{B}^0\boldsymbol{u}(k)] \\ &\quad \oplus \boldsymbol{A}_{k-1}^0\boldsymbol{x}(k-1) \oplus \boldsymbol{B}^0\boldsymbol{u}(k) \\ &= \boldsymbol{F}_k^2\boldsymbol{x}(k) \oplus (\boldsymbol{e}_n \oplus \boldsymbol{F}_k)\boldsymbol{A}_{k-1}^0\boldsymbol{x}(k-1) \\ &\quad \oplus (\boldsymbol{e}_n \oplus \boldsymbol{F}_k)\boldsymbol{B}^0\boldsymbol{u}(k)\end{aligned} \tag{19}$$

Repeating the same procedure produces the next representation:

$$\boldsymbol{x}(k) = \boldsymbol{F}_k^l\boldsymbol{x}(k) \oplus \boldsymbol{F}_k^*[\boldsymbol{A}_{k-1}^0\boldsymbol{x}(k-1) \oplus \boldsymbol{B}^0\boldsymbol{u}(k)] \tag{20}$$

where $\boldsymbol{F}_k^* = \boldsymbol{e}_n \oplus \boldsymbol{F}_k \oplus \boldsymbol{F}_k^2 \oplus \cdots \oplus \boldsymbol{F}_k^{l-1}$. With the help of theorem 1, there is an instance of l $(1 \le l \le n)$ that satisfies $\boldsymbol{F}_k^l = \boldsymbol{\varepsilon}_n$. Hence, the first term in the right hand side of equation (20) is eliminated, which means that it is transformed into the form of equation (1). Thus, it reveals that the following relations hold:

$$\boldsymbol{A} = \boldsymbol{F}_k^*\boldsymbol{A}_{k-1}^0, \quad \boldsymbol{B} = \boldsymbol{F}_k^*\boldsymbol{B}^0. \tag{21}$$

Let us consider a case for Figure 1 as an example. Recalling that the definitions of the relevant matrices are determined as follows:

$$\boldsymbol{A}_{k-1}^0 = \begin{pmatrix} d_1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & d_2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & d_3 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & d_4 \end{pmatrix}, \quad \boldsymbol{F}_k = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ d_1 & \varepsilon & \varepsilon & \varepsilon \\ d_1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & d_2 & d_3 & \varepsilon \end{pmatrix},$$

$$\boldsymbol{B}^0 = \begin{pmatrix} e & \varepsilon & \varepsilon & \varepsilon \end{pmatrix}^T.$$

For the adjacency matrix, the following relations are satisfied:

$$\boldsymbol{F}_k^2 = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ f_{41}^2 & \varepsilon & \varepsilon & \varepsilon \end{pmatrix}, \quad \boldsymbol{F}_k^3 = \boldsymbol{\varepsilon}_{44}, \quad l = 3 \le n = 4,$$

$$\boldsymbol{F}_k^* = \begin{pmatrix} e & \varepsilon & \varepsilon & \varepsilon \\ d_1 & e & \varepsilon & \varepsilon \\ d_1 & \varepsilon & e & \varepsilon \\ f_{41}^* & d_2 & d_3 & e \end{pmatrix}, \quad f_{41}^2 = f_{41}^* = d_1(d_2 \oplus d_3).$$

Subsequently, multiplying $\boldsymbol{F}_k^*$ by $\boldsymbol{A}_{k-1}^0$ and $\boldsymbol{B}^0$ leads to $\boldsymbol{A}$ and $\boldsymbol{B}$ in equation (21), which are coincident with those in equation (8).

As these matters indicate, the behavior of systems whose structures satisfy the conditions assumed in the beginning of this section can be described in MPL form shown in equations (1) and (2).

4. SCHEDULING METHOD

This section derives the earliest/latest starting times in entire processes, calculates the float times, and proposes a method for finding bottlenecks. In subsequent discussions, the earliest starting times in all processes are denoted by $\boldsymbol{x}_E$, those for the latest starting time by $\boldsymbol{x}_L$, and description of the event counter (k) is often abbreviated for simplicity.

4.1 Earliest Starting Time

The earliest starting time is the minimum value on which the corresponding process can start manufacturing immediately. For process $i\,(1 \le i \le n)$, it is stated as:

$$\begin{aligned}ET_i = [\boldsymbol{x}_E]_i &= \bigoplus_{j \in \mathcal{P}_i}[\boldsymbol{u}]_j \oplus \bigoplus_{j \in \mathcal{R}_i}([\boldsymbol{x}_E]_j + d_j) \\ &\quad \oplus ([\boldsymbol{x}(k-1)]_i + d_i(k-1))\end{aligned}, \tag{22}$$

where $\mathcal{P}_i$ and $\mathcal{R}_i$ are the same variables used in equations (11) and (12). The first term in the right hand side represents the maximum value among the feeding times from external outputs, and set ε if $\mathcal{P}_i = \{\phi\}$. The second term indicates the maximum value among the completion times in preceding processes, and set ε in case of $\mathcal{R}_i = \{\phi\}$. Moreover, the third term represents the completion time of the previous batch. Recalling equations (10)-(12), the next relations can be used:

$$\bigoplus_{j \in \mathcal{P}_i}[\boldsymbol{u}]_j = [\boldsymbol{B}^0\boldsymbol{u}]_i, \quad \bigoplus_{j \in \mathcal{R}_i}([\boldsymbol{x}_E]_j + d_j) = [\boldsymbol{F}\boldsymbol{x}_E]_i,$$

$$[\boldsymbol{x}(k-1)]_i + d_i(k-1) = [\boldsymbol{A}_{k-1}^0 \boldsymbol{x}(k-1)]_j .$$

Note the subscript k for $\boldsymbol{F}_k$ is abbreviated. Hence, the earliest starting time of process i $(1 \le i \le n)$, $[\boldsymbol{x}_E]_i$, can be summarized as shown below:

$$[\boldsymbol{x}_E]_i = [\boldsymbol{B}^0 \boldsymbol{u}]_i \oplus [\boldsymbol{F}\boldsymbol{x}_E]_i \oplus [\boldsymbol{A}_{k-1}^0 \boldsymbol{x}(k-1)]_i . \tag{23}$$

The above equation holds the same form as equation (15), which implies that it is also equivalent to equation (1) by substituting $\boldsymbol{x}(k)$ by $\boldsymbol{x}_E$. Accordingly, the earliest starting times of any of the processes are summarized as:

$$\boldsymbol{x}_E = \boldsymbol{F}^* [\boldsymbol{A}_{k-1}^0 \boldsymbol{x}(k-1) \oplus \boldsymbol{B}^0 \boldsymbol{u}] . \tag{24}$$

Moreover, the corresponding output times is given by:

$$\boldsymbol{y} = \boldsymbol{C}_k \boldsymbol{x}_E . \tag{25}$$

with the help of equation (2).

4.2 Latest Starting Time

This paper defines the latest starting time as the maximum value for which the same output time by the earliest time is accomplished. Thus, the latest starting time in process i $(1 \le i \le n)$ can be expressed as follows:

$$LT_i = [\boldsymbol{x}_L]_i = (\textstyle\bigwedge_{j \in \mathcal{Q}_i} [\boldsymbol{y}]_j) \wedge (\textstyle\bigwedge_{j \in S_i} [\boldsymbol{x}_L]_j) - d_i . \tag{26}$$

where $\mathcal{Q}_i$ and S_i represent a number set of external outputs attached to process i and a number set of succeeding processes, respectively. This paper handles systems whose structures satisfy the following relation:

$$S_i = \{\phi\} \Rightarrow \mathcal{Q}_i \ne \{\phi\} . \tag{27}$$

Note that there may possibly be processes that have both succeeding processes and external outputs, which means that the inverse of equation (27) does not hold. The first term of the right hand side in equation (26) indicates the minimum value among the output times of external outputs, and set $-\varepsilon$ if $\mathcal{Q}_i = \{\phi\}$. The second term represents the minimum value among the starting times in succeeding processes, and set $-\varepsilon$ in case of $S_i = \{\phi\}$. Equation (27) implies that $[\boldsymbol{x}_L]_i$ takes finite values for all i $(1 \le i \le n)$. On the other hand, equation (26) can be transformed in the following way:

$$\begin{aligned}
[\boldsymbol{x}_L]_i &= (\textstyle\bigwedge_{j \in \mathcal{Q}_i} [\boldsymbol{y}]_j - d_i) \wedge (\textstyle\bigwedge_{j \in S_i} [\boldsymbol{x}_L]_j - d_i) \\
&= (\textstyle\bigwedge_{j \in \mathcal{Q}_i} [\boldsymbol{y}]_j - [\boldsymbol{C}]_{ji}) \wedge (\textstyle\bigwedge_{j \in S_i} [\boldsymbol{x}_L]_j - [\boldsymbol{F}]_{ji}) \\
&= (\textstyle\bigwedge_{j=1}^{q} [\boldsymbol{C}^T]_{ij} \setminus [\boldsymbol{y}]_j) \wedge (\textstyle\bigwedge_{j=1}^{n} [\boldsymbol{F}^T]_{ij} \setminus [\boldsymbol{x}_L]_j) \\
&= [\boldsymbol{C}^T \odot \boldsymbol{y}]_i \wedge [\boldsymbol{F}^T \odot \boldsymbol{x}_L]_i
\end{aligned}$$

Since this relation holds for all i $(1 \le i \le n)$, the latest starting times for the whole processes can be expressed as:

$$\boldsymbol{x}_L = (\boldsymbol{C}^T \odot \boldsymbol{y}) \wedge (\boldsymbol{F}^T \odot \boldsymbol{x}_L) . \tag{28}$$

Equation (28) includes $\boldsymbol{x}_L$ itself in the right hand side. Hence, the next three identical equations are used to transfer to a simpler form.

Theorem 2. In $\boldsymbol{X}, \boldsymbol{Y} \in \mathcal{D}^{l \times m}$, $\boldsymbol{Z} \in \mathcal{D}^{m \times l}$, $\boldsymbol{v}, \boldsymbol{w} \in \mathcal{D}^m$, the following identical equation holds.

$$(\boldsymbol{X} \oplus \boldsymbol{Y}) \odot \boldsymbol{v} = (\boldsymbol{X} \odot \boldsymbol{v}) \wedge (\boldsymbol{Y} \odot \boldsymbol{v}), \tag{29}$$

$$\boldsymbol{X}^T \odot (\boldsymbol{Z}^T \odot \boldsymbol{v}) = (\boldsymbol{Z}\boldsymbol{X})^T \odot \boldsymbol{v}, \tag{30}$$

$$\boldsymbol{X} \odot (\boldsymbol{v} \wedge \boldsymbol{w}) = (\boldsymbol{X} \odot \boldsymbol{v}) \wedge (\boldsymbol{X} \odot \boldsymbol{w}). \tag{31}$$

Proof. Equation (29) is proved in (Masuda *et al.*, 2003). Regarding equations (30) and (31), Cohen *et al.*, (1989) summarized for a special case in $l = m = n = 1$. We now prove them for matrices. The i th element of the left hand side in equation (30) can be calculated as:

$$\begin{aligned}
&[\boldsymbol{X}^T \odot (\boldsymbol{Z}^T \odot \boldsymbol{v})]_i \\
&= \bigwedge_{k=1}^{l} \{[\boldsymbol{X}^T]_{ik} \setminus \bigwedge_{j=1}^{m} ([\boldsymbol{Z}^T]_{kj} \setminus [\boldsymbol{v}]_j)\} . \\
&= \bigwedge_{k=1}^{l} \bigwedge_{j=1}^{m} (-[\boldsymbol{X}]_{ki} - [\boldsymbol{Z}]_{jk} + [\boldsymbol{v}]_j)
\end{aligned} \tag{32}$$

On the other hand, the i th element of the right hand side can be expanded as shown below:

$$\begin{aligned}
[(\boldsymbol{Z}\boldsymbol{X})^T \odot \boldsymbol{v}]_i &= \bigwedge_{j=1}^{m} \{[\bigoplus_{k=1}^{l} ([\boldsymbol{Z}]_{ik} + [\boldsymbol{X}]_{kj})^T]_{ij} \setminus [\boldsymbol{v}]_j\} \\
&= \bigwedge_{j=1}^{m} [\bigoplus_{k=1}^{l} ([\boldsymbol{Z}]_{jk} + [\boldsymbol{X}]_{ki}) \setminus [\boldsymbol{v}]_j] . \\
&= \bigwedge_{j=1}^{m} \bigwedge_{k=1}^{l} (-[\boldsymbol{Z}]_{jk} - [\boldsymbol{X}]_{ki} + [\boldsymbol{v}]_j)
\end{aligned} \tag{33}$$

Since the orders of the operators $\wedge$ are changeable, equations (32) and (33) are equal for all i $(1 \le i \le m)$, thus equation (30) is proved. Next, the i th element of equation (31) can be transformed as:

$$\begin{aligned}[\boldsymbol{X}\odot(\boldsymbol{v}\wedge\boldsymbol{w})]_i &= \bigwedge_{j=1}^{n}\{[\boldsymbol{X}]_{ij}\setminus([\boldsymbol{v}]_j\wedge[\boldsymbol{w}]_j)\}\\ &= \bigwedge_{j=1}^{n}\{-[\boldsymbol{X}]_{ij}+([\boldsymbol{v}]_j\wedge[\boldsymbol{w}]_j)\}\\ &= \{\bigwedge_{j=1}^{n}(-[\boldsymbol{X}]_{ij}+[\boldsymbol{v}]_j)\}\wedge\{\bigwedge_{j=1}^{n}\cdot\\ &\quad(-[\boldsymbol{X}]_{ij}+[\boldsymbol{w}]_j)\}\\ &= [\boldsymbol{X}\odot\boldsymbol{v}]_i\wedge[\boldsymbol{X}\odot\boldsymbol{w}]_i\end{aligned}$$

The above equality holds for all i $(1\le i\le n)$, which implies that equation (31) is proved. □

Utilizing equations (30) and (31), operating $\boldsymbol{F}^T\odot$ on both hand sides of equation (28) leads to next equation:

$$\begin{aligned}\boldsymbol{F}^T\odot\boldsymbol{x}_L &= \boldsymbol{F}^T\odot[(\boldsymbol{C}^T\odot\boldsymbol{y})\wedge(\boldsymbol{F}^T\odot\boldsymbol{x}_L)]\\ &= [(\boldsymbol{CF})^T\odot\boldsymbol{y}]\wedge[\boldsymbol{F}^{2^T}\odot\boldsymbol{x}_L]\end{aligned} \tag{34}$$

Similarly, operating $\boldsymbol{F}^{i^T}\odot$ $(2\le i\le l-1)$ on both hand sides, the following equations are obtained:

$$\begin{aligned}\boldsymbol{F}^{2^T}\odot\boldsymbol{x}_L &= [(\boldsymbol{CF}^2)^T\odot\boldsymbol{y}]\wedge[\boldsymbol{F}^{3^T}\odot\boldsymbol{x}_L]\\ &\vdots\\ \boldsymbol{F}^{l-2^T}\odot\boldsymbol{x}_L &= [(\boldsymbol{CF}^{l-2})^T\odot\boldsymbol{y}]\wedge[\boldsymbol{F}^{l-1^T}\odot\boldsymbol{x}_L]\\ \boldsymbol{F}^{l-1^T}\odot\boldsymbol{x}_L &= [(\boldsymbol{CF}^{l-1})^T\odot\boldsymbol{y}]\end{aligned} \tag{35}$$

By substituting sequentially using equations (28)-(30), (34), and (35), the latest starting time can be summarized in the following way:

$$\begin{aligned}\boldsymbol{x}_L &= (\boldsymbol{C}^T\odot\boldsymbol{y})\wedge[(\boldsymbol{CF})^T\odot\boldsymbol{y}]\wedge\cdots\\ &\quad\wedge[(\boldsymbol{CF}^{l-1})^T\odot\boldsymbol{y}]\\ &= [\boldsymbol{e}_n\odot(\boldsymbol{C}^T\odot\boldsymbol{y})]\wedge[\boldsymbol{F}^T\odot(\boldsymbol{C}^T\odot\boldsymbol{y})]\\ &\quad\wedge\cdots\wedge[\boldsymbol{F}^{l-1^T}\odot(\boldsymbol{C}^T\odot\boldsymbol{y})]\\ &= (\boldsymbol{e}_n\oplus\boldsymbol{F}\oplus\cdots\oplus\boldsymbol{F}^{l-1})^T\odot(\boldsymbol{C}^T\odot\boldsymbol{y})\\ &= \boldsymbol{F}^{*^T}\odot(\boldsymbol{C}^T\odot\boldsymbol{y})=(\boldsymbol{CF}^*)^T\odot\boldsymbol{y}\end{aligned}. \tag{36}$$

Moreover, equation (25) leads the next representation:

$$\boldsymbol{x}_L=(\boldsymbol{CF}^*)^T\odot\boldsymbol{y}=(\boldsymbol{CF}^*)^T\odot(\boldsymbol{Cx}_E). \tag{37}$$

As described above, the latest starting times for all processes are obtained.
The fact that equation (37) gives the latest starting times and they are unique is also confirmed by considering the properties of the greatest subsolution (Cohen *et al.*, 1989; Baccelli *et al.*, 1992). The greatest subsolution is a method for solving linear equation in max-plus algebra. Let us consider solving a linear equation $\boldsymbol{M}\otimes\boldsymbol{z}=\boldsymbol{v}$. By relaxing this to an inequality, the maximum solution of each element that satisfies the inequality is equal to the greatest subsolution. The specific solution is obtained utilizing the Residuation theory. It is:

$$\bar{\boldsymbol{z}}=\max\{[\boldsymbol{z}]_i\,|\,[\boldsymbol{M}\otimes\boldsymbol{z}]_i\le[\boldsymbol{v}]_i\}=\boldsymbol{M}^T\odot\boldsymbol{v}. \tag{38}$$

Therefore, it revealed that $\boldsymbol{x}_L$ in equation (37) is the greatest subsolution for the next linear equation:

$$(\boldsymbol{CF}^*)\boldsymbol{x}=\boldsymbol{Cx}_E=\boldsymbol{y}. \tag{39}$$

Two fundamental properties of the greatest subsolution are reviewed here; the first issue is the uniqueness of the solution. Secondly, for all $\bar{\boldsymbol{z}}_+$ in which there are one or multiple elements greater than $\bar{\boldsymbol{z}}$, the order relation in equation (38) is no longer held, and there is an instance i which satisfies $[\boldsymbol{M}\otimes\bar{\boldsymbol{z}}_+]_i>[\boldsymbol{v}]_i$ (Goto and Masuda, 2004b). This means that the output time is delayed. Hence, all elements of $\boldsymbol{x}$ in equation (39) must be equal or less than the greatest subsolution, which indicates that equation (37) is equivalent to the latest starting time.

Furthermore, the latest feeding time $\boldsymbol{u}_L$ that provides the same output time $\boldsymbol{y}$ is considered. It is equal to the minimum value among the latest starting times in processes that have the corresponding external input. It can be calculated as:

$$\begin{aligned}[\boldsymbol{u}_L]_i &= \bigwedge_{j\in\mathcal{W}_i}[\boldsymbol{x}_L]_j=(\bigwedge_{j\in\mathcal{W}_i}[\boldsymbol{x}_L]_j-[\boldsymbol{B}^0]_{ji})\\ &= (\wedge_{j=1}^{n}[\boldsymbol{B}^{0^T}]_{ij}\setminus[\boldsymbol{x}_L]_j)=[\boldsymbol{B}^{0^T}\odot\boldsymbol{x}_L]_i\end{aligned},$$

where $\mathcal{W}_i$ represents the number set of succeeding processes attached to external input i. The above equation holds for all $i\,(1\le i\le p)$, and recalling equations (30) and (36) leads to the following representation:

$$\begin{aligned}\boldsymbol{u}_L=\boldsymbol{B}^{0^T}\odot\boldsymbol{x}_L &= \boldsymbol{B}^{0^T}\odot[(\boldsymbol{CF}^*)^T\odot\boldsymbol{y}]\\ &= (\boldsymbol{CF}^*\boldsymbol{B}^0)^T\odot\boldsymbol{y}=(\boldsymbol{CB})^T\odot\boldsymbol{y}\end{aligned}$$

This result is consistent with those in references Cohen *et al.* (1989) and Masuda *et al.* (2003) in which they consider the relation between the inputs and outputs of the system.

4.3 Total Float and Bottleneck Process

This subsection finds bottleneck processes utilizing the earliest starting time $\boldsymbol{x}_E$ and the latest starting time $\boldsymbol{x}_L$ derived in the previous subsections.

A Bottleneck process is defined as one whose total float is zero. Moreover, total float is defined as total sum of float times for the corresponding process. It is also stated as the difference between two fundamental times; one is the minimum value among the latest starting times of the succeeding processes by which the output time is invariant, and the other is the completion time in the corresponding process caused by the earliest starting time. This is formulated as follows:

$$TF_i = (\bigwedge_{j \in \boldsymbol{S}_i} [\boldsymbol{x}_L]_j) \wedge (\bigwedge_{j \in \boldsymbol{\mathcal{Q}}_i} [\boldsymbol{y}]_j) - ([\boldsymbol{x}_E]_i + d_i), \quad (40)$$

where $\boldsymbol{S}_i$ and $\boldsymbol{\mathcal{Q}}_i$ are the same variables as used in equation (26). The first term of the right hand side expresses the constraints caused by succeeding processes, and the second term represents the constraints of output times for external outputs. Set $-\varepsilon$ to each term if $\boldsymbol{S}_i = \{\phi\}$ or $\boldsymbol{\mathcal{Q}}_i = \{\phi\}$. Utilizing equation (27) reveals that the total float TF_i takes a finite value for all i.

Let us represent TF_i in equation (40) in a simpler form. First, the following transformation is performed:

$$\begin{aligned} TF_i &= (\wedge_{j \in \boldsymbol{S}_i} [\boldsymbol{x}_L]_j - d_i) \wedge \\ &\quad (\wedge_{j \in \boldsymbol{\mathcal{Q}}_i} [\boldsymbol{y}]_j - d_i) - [\boldsymbol{x}_E]_i \\ &= (\wedge_{j \in \boldsymbol{S}_i} [\boldsymbol{x}_L]_j - [\boldsymbol{F}]_{ji}) \wedge \\ &\quad (\wedge_{j \in \boldsymbol{\mathcal{Q}}_i} [\boldsymbol{y}]_j - [\boldsymbol{C}]_{ji}) - [\boldsymbol{x}_E]_i \qquad (41) \\ &= (\wedge_{j=1}^{n} [\boldsymbol{F}^T]_{ij} \backslash [\boldsymbol{x}_L]_j) \wedge \\ &\quad (\wedge_{j=1}^{q} [\boldsymbol{C}^T]_{ij} \backslash [\boldsymbol{y}]_j) - [\boldsymbol{x}_E]_i \\ &= [\boldsymbol{F}^T \odot \boldsymbol{x}_L]_i \wedge [\boldsymbol{C}^T \odot \boldsymbol{y}]_i - [\boldsymbol{x}_E]_i \end{aligned}$$

The above equation holds true for all i $(1 \le i \le n)$. Hence, denoting the total floats of all processes by $\boldsymbol{w}$, it can be simply expressed as $\boldsymbol{w} = (\boldsymbol{F}^T \odot \boldsymbol{x}_L) \wedge (\boldsymbol{C}^T \odot \boldsymbol{y}) - \boldsymbol{x}_E$. On the other hand, utilizing equations (28), (30) and (31), the first term of the right hand side in equation (41) can be simplified to:

$$\begin{aligned} &(\boldsymbol{F}^T \odot \boldsymbol{x}_L) \wedge (\boldsymbol{C}^T \odot \boldsymbol{y}) \\ &\quad = \{\boldsymbol{F}^T \odot [(\boldsymbol{C}\boldsymbol{F}^*)^T \odot \boldsymbol{y})]\} \wedge (\boldsymbol{C}^T \odot \boldsymbol{y}) \\ &\quad = [\boldsymbol{F}_+^{*T} \odot (\boldsymbol{C}^T \odot \boldsymbol{y})] \wedge [\boldsymbol{e}_n^T \odot (\boldsymbol{C}^T \odot \boldsymbol{y})], \\ &\quad = (\boldsymbol{F}_+^* \oplus \boldsymbol{e}_n)^T \odot (\boldsymbol{C}^T \odot \boldsymbol{y}) = \boldsymbol{F}^{*T} \odot (\boldsymbol{C}^T \odot \boldsymbol{y}) \\ &\quad = (\boldsymbol{C}\boldsymbol{F}^*)^T \odot \boldsymbol{y} = \boldsymbol{x}_L \end{aligned}$$

where $\boldsymbol{F}_+^* = \boldsymbol{F} \oplus \boldsymbol{F}^2 \oplus \cdots \oplus \boldsymbol{F}^{l-1}$ satisfies $\boldsymbol{F}^* \boldsymbol{F} = \boldsymbol{F}_+^*$ and $\boldsymbol{F}_+^* \oplus \boldsymbol{e}_n = \boldsymbol{F}^*$. Consequently, the total floats regarding all processes $\boldsymbol{w}$ are obtained as:

$$\boldsymbol{w} = \boldsymbol{x}_L - \boldsymbol{x}_E = (\boldsymbol{C}\boldsymbol{F}^*)^T \odot (\boldsymbol{C}\boldsymbol{x}_E) - \boldsymbol{x}_E.$$

The bottleneck processes are given by the collection of the process numbers i that satisfy the next equation:

$$\text{Bottlenecks: } \{i \mid [\boldsymbol{w}]_i = 0\}. \qquad (42)$$

4.4 Rescheduling

In the previous discussions an ideal situation is assumed; the operation is steadily performed according to the initial plan where all relevant parameters such as feeding times from external inputs or processing times in the respective processes never vary. However, in practice, several changes that could influence the output time $\boldsymbol{y}$ often occur after the processing has started. Typical cases are:

- Arrival times of material: influences the feeding times $\boldsymbol{u}$.
- Processing times: influence the system parameters d that effect on the system matrices $\boldsymbol{A}_{k-1}^0$, $\boldsymbol{C}_k$, and $\boldsymbol{F}_k$.
- Starting times of processing: influence the state variables $\boldsymbol{x}$.

Since these changes could vary the initial plan such as the earliest/latest starting times and/or bottleneck processes, it is important to develop an efficient rescheduling method in the field of schedule control and project management.

Suppose the feeding time, processing time and/or starting time changes for some reason after the processing has started. We append symbols $[\tilde{\ }]$ for representing the changed variables as:

$$[\boldsymbol{u}]_i \to [\tilde{\boldsymbol{u}}]_i \ (1 \le i \le p), \ [\boldsymbol{x}]_i \to [\tilde{\boldsymbol{x}}]_i \ (1 \le i \le n).$$

In a similar way, we express the system, output, and adjacency matrices that may be influenced by the changes of the processing times as:

$$\boldsymbol{A}_{k-1}^0 \to \tilde{\boldsymbol{A}}_{k-1}^0, \ \boldsymbol{C} \to \tilde{\boldsymbol{C}}, \ \boldsymbol{F} \to \tilde{\boldsymbol{F}}. \qquad (43)$$

Regarding the input matrix $\boldsymbol{B}^0$, it remains unchanged recalling its definition. The updated state variables are recalculated using these matrices. First, set $[\tilde{\boldsymbol{x}}^{(0)}]_i = \varepsilon$ for all i that are to be recalculated. Processes whose starting times are not to be recalculated are those where the processing has already started, or those where the starting time has changed and fixed due to delay. In a similar way to equation (23), the earliest starting time in the most upstream process $[\tilde{\boldsymbol{x}}^{(1)}]_i$ among targets for recalculation can be represented as follows:

$$[\tilde{\boldsymbol{x}}^{(1)}]_i = [\boldsymbol{B}^0 \tilde{\boldsymbol{u}}]_i \oplus [\tilde{\boldsymbol{F}} \tilde{\boldsymbol{x}}^{(0)}]_i \oplus [\tilde{\boldsymbol{A}}_{k-1}^0 \boldsymbol{x}(k-1)]_i . \quad (44)$$

The meaning of the 'most upstream' represents a process whose all preceding processes are not to be recalculated, and there may be multiple ones. Repeating the same transformation in equation (19), the earliest starting times in succeeding processes are obtained iteratively as follows:

$$\begin{aligned} \tilde{\boldsymbol{x}}^{(2)} &= \boldsymbol{B}^0 \tilde{\boldsymbol{u}} \oplus \tilde{\boldsymbol{F}} \tilde{\boldsymbol{x}}^{(1)} \oplus \tilde{\boldsymbol{A}}_{k-1}^0 \boldsymbol{x}(k) = \tilde{\boldsymbol{F}}^2 \tilde{\boldsymbol{x}}^{(0)} \\ &\quad \oplus (\boldsymbol{e}_n \oplus \tilde{\boldsymbol{F}})[\tilde{\boldsymbol{A}}_{k-1}^0 \boldsymbol{x}(k-1) \oplus \boldsymbol{B}^0 \tilde{\boldsymbol{u}})] \\ &\vdots \\ \tilde{\boldsymbol{x}}^{(l-1)} &= \boldsymbol{B}^0 \tilde{\boldsymbol{u}} \oplus \tilde{\boldsymbol{F}} \tilde{\boldsymbol{x}}^{(l-2)} \oplus \tilde{\boldsymbol{A}}_{k-1}^0 \boldsymbol{x}(k-1) \\ &= \tilde{\boldsymbol{F}}^{l-1} \boldsymbol{x}^{(0)} \oplus (\boldsymbol{e}_n \oplus \tilde{\boldsymbol{F}} \oplus \cdots \oplus \tilde{\boldsymbol{F}}^{l-2}) , \\ &\quad \otimes [\tilde{\boldsymbol{A}}_{k-1}^0 \boldsymbol{x}(k-1) \oplus \boldsymbol{B}^0 \tilde{\boldsymbol{u}}] \\ \tilde{\boldsymbol{x}}^{(l)} &= \boldsymbol{B}^0 \tilde{\boldsymbol{u}} \oplus \tilde{\boldsymbol{F}} \tilde{\boldsymbol{x}}^{(l-1)} \oplus \tilde{\boldsymbol{A}}_{k-1}^0 \boldsymbol{x}(k-1) \\ &= \tilde{\boldsymbol{F}}^* [\tilde{\boldsymbol{A}}_{k-1}^0 \boldsymbol{x}(k-1) \oplus \boldsymbol{B}^0 \tilde{\boldsymbol{u}}] \end{aligned} \quad (45)$$

where $\tilde{\boldsymbol{F}}^* = \tilde{\boldsymbol{F}}_k^* = \boldsymbol{e}_n \oplus \tilde{\boldsymbol{F}} \oplus \cdots \oplus \tilde{\boldsymbol{F}}^{l-1}$. Note that the statement of the element number $[\cdot]_i$ is abbreviated here. The updated earliest starting times $\tilde{\boldsymbol{x}}_E$ are obtained as:

$$\tilde{\boldsymbol{x}}_E = \tilde{\boldsymbol{x}}^{(0)} \oplus \tilde{\boldsymbol{x}}^{(1)} \oplus \cdots \oplus \tilde{\boldsymbol{x}}^{(l)} .$$

Utilizing equations (44) and (45), it is simply expressed as:

$$\tilde{\boldsymbol{x}}_E = \tilde{\boldsymbol{F}}^* \tilde{\boldsymbol{x}}^{(0)} \oplus \tilde{\boldsymbol{F}}^* [\tilde{\boldsymbol{A}}_{k-1}^0 \boldsymbol{x}(k-1) \oplus \boldsymbol{B}^0 \tilde{\boldsymbol{u}}] . \quad (46)$$

Equation (46) is a general form of equation (24), which is applicable even when the related parameters change after the processing has started. If the present time is before the starting time, it means $[\tilde{\boldsymbol{x}}^{(0)}]_i = \varepsilon$ for all i $(1 \le i \le n)$ and indicates that it is equivalent to equation (24).

Consider a special case where the starting time of processing has delayed for some reason. By replacing the elements in which the delay of the starting time has been confirmed by the newest values, $[\tilde{\boldsymbol{x}}^{(0)}]_i \ge [\boldsymbol{x}_E]_i$ is followed for all $i (1 \le i \le n)$. Since the system matrices in equation (43) are invariant, the next relation holds:

$$\begin{aligned} \tilde{\boldsymbol{x}}^{(0)} \ge \boldsymbol{x}_E &= \boldsymbol{F}^* [\boldsymbol{A}_{k-1}^0 \boldsymbol{x}(k-1) \oplus \boldsymbol{B}^0 \boldsymbol{u}] \\ &= \tilde{\boldsymbol{F}}^* [\tilde{\boldsymbol{A}}_{k-1}^0 \boldsymbol{x}(k-1) \oplus \boldsymbol{B}^0 \tilde{\boldsymbol{u}}] \end{aligned} .$$

Therefore, in this case, equation (46) can be expressed simply as follows:

$$\tilde{\boldsymbol{x}}_E = \boldsymbol{F}^* \tilde{\boldsymbol{x}}^{(0)} . \quad (47)$$

The corresponding output times, latest starting times, and bottlenecks can be found and obtained in analogous ways to equations (25), (36), and (42):

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{C}} \tilde{\boldsymbol{x}}_E , \quad \tilde{\boldsymbol{x}}_L = (\tilde{\boldsymbol{C}} \tilde{\boldsymbol{F}}^*)^T \odot \tilde{\boldsymbol{y}} ,$$

$$\text{Bottlenecks: } \{ i \mid [\tilde{\boldsymbol{x}}_L]_i - [\tilde{\boldsymbol{x}}_E]_i = 0 \} .$$

Bottleneck processes are principally based on the precedence constraints and the processing times of the corresponding system. However, if the related parameters change after the processing has started, its locations could possibly vary accordingly. The next section verifies this.

5. NUMERICAL EXAMPLE

This section demonstrates examples of the methods for calculating the earliest/latest starting times and finding the bottlenecks introduced in the previous section utilizing a simple production system with precedence constraints.

Table 1 shows the parameters imposed on a certain production system. The precedence constraints and locations of inputs/outputs for this system are illustrated in Figure 2. The elements of the matrices in equations (10)-(13) are as follows:

$$\boldsymbol{A}_{k-1}^0 = \mathrm{diag}[1, 6, 2, 3, 4] , \quad \boldsymbol{F} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon \\ \varepsilon & 6 & \varepsilon & 3 & \varepsilon \end{bmatrix} ,$$

$$\boldsymbol{B}^0 = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \end{bmatrix}^T , \quad \boldsymbol{C} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4 \end{bmatrix} .$$

By calculating $\boldsymbol{F}^2$, $\boldsymbol{F}^3, \cdots$ iteratively, we obtain $\boldsymbol{F}^4 = \boldsymbol{\varepsilon}_{55}$ and the following relation:

$$\boldsymbol{F}^* = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1 & e & \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & e & \varepsilon & \varepsilon \\ 3 & \varepsilon & 2 & e & \varepsilon \\ 7 & 6 & 5 & 3 & e \end{bmatrix} .$$

Supposing the initial condition as $\boldsymbol{x}(k) = \boldsymbol{\varepsilon}_{51}$ and input times from external input as $\boldsymbol{u} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$, the earliest starting time and the output time can be calculated in the following way:

$$\begin{aligned}\boldsymbol{x}_E &= \boldsymbol{F}^*\boldsymbol{A}^0_{k-1}\boldsymbol{x}(k-1) \oplus \boldsymbol{F}^*\boldsymbol{B}^0\boldsymbol{u} \\ &= \boldsymbol{F}^*\boldsymbol{B}^0\boldsymbol{u} = [0 \quad 1 \quad 1 \quad 3 \quad 7]^T\end{aligned}, \quad \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}_E = 11 .$$

Next, the latest starting time, feeding time and total float are obtained as follows:

$$\begin{aligned}\boldsymbol{x}_L &= (\boldsymbol{C}\boldsymbol{F}^*)^T \odot \boldsymbol{y} = [0 \ 1 \ 2 \ 4 \ 7]^T \\ \boldsymbol{u}_L &= (\boldsymbol{C}\boldsymbol{F}^*\boldsymbol{B}^0)^T \odot \boldsymbol{y} = [0 \ 2]^T \\ \boldsymbol{w} &= \boldsymbol{x}_L - \boldsymbol{x}_E = [0 \ 0 \ 1 \ 1 \ 0]^T\end{aligned} .$$

Therefore, the bottleneck processes can be identified as $\{1, 2, 5\}$. Figure 3 illustrates the plan for the resource employment in each process using a Gantt chart.

Subsequently, in an experiment for tracing a change of the earliest/latest time and the location of bottlenecks when the related parameters have changed, assume that the starting time for manufacturing in process 4 has changed to $[\tilde{\boldsymbol{x}}^{(0)}]_4 = 5$ on $t = 4$. If all of the processing times are invariant, the updated earliest starting time and output times can be calculated using equation (47):

$$\begin{aligned}\tilde{\boldsymbol{x}}_E = \boldsymbol{F}^*\tilde{\boldsymbol{x}}^{(0)} &= \boldsymbol{F}^*[0 \quad 1 \quad 1 \quad \underline{5} \quad 7]^T \\ &= [0 \quad 1 \quad 1 \quad 5 \quad 8]^T\end{aligned}, \quad \tilde{\boldsymbol{y}} = \boldsymbol{C}\tilde{\boldsymbol{x}}^{(0)} = 12 .$$

The underlined element is the delayed starting time. It is delayed for two unit times from the earliest starting time. However, since process 4 is not the bottleneck process, its completion time is delayed for only one unit time. The updated latest starting time and the total float are calculated as follows:

$$\begin{aligned}\tilde{\boldsymbol{x}}_L &= (\boldsymbol{C}\boldsymbol{F}^*)^T \odot \tilde{\boldsymbol{y}} = [1 \quad 2 \quad 3 \quad 5 \quad 8]^T \\ \boldsymbol{w} &= \tilde{\boldsymbol{x}}_L - \tilde{\boldsymbol{x}}_E = [1 \quad 1 \quad 2 \quad 0 \quad 0]^T\end{aligned} .$$

This reveals that the location of the bottleneck has moved to $\{4, 5\}$, which implies that the delay in process 4 is not allowed whereas the completion time in process 2 can be delayed for one unit of time.

As the above discussions imply, the methods proposed in this paper provide solutions for the following problems: scheduling problem for a production system with precedence constraints, process planning and finding bottlenecks, and rescheduling when a delay caused by an unpredicted reason has occurred after the processing has started.

Table 1. Parameters of a manufacturing system.

Process No.	Processing time	Preceding processes	Input No.	Output No.
(1)	1	--	1	--
(2)	6	(1)	--	--
(3)	2	(1)	2	--
(4)	3	(3)	--	--
(5)	4	(2), (4)	--	1

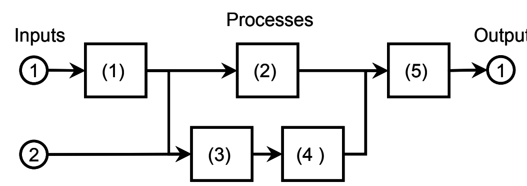

Figure 2. A manufacturing process with two-inputs and one-output

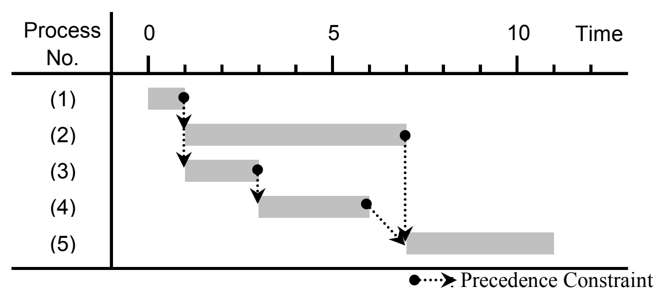

Figure 3. A Gantt chart of an initial schedule

6. CONCLUSION

This paper proposed an approach to monitoring and scheduling methods for a class of repetitive MIMO-FIFO systems, based on MPL representation. Though we presented examples for production systems, the methods can also be applied to problems in project management. Since they are applicable to MIMO systems, they can be applied to wider scope than those based on PERT.

Initially we considered a process for representing the systems behavior in MPL form and derived equations taking into account precedence constraints, processing times, locations of external inputs and outputs. Next, the earliest and latest starting times, and total floats were formulated from their original definitions based on max and min operations. These were transformed into simpler MPL equations, and the locations of bottleneck processes were identified.

Later, we proposed a rescheduling method for cases where relevant parameters such as starting times for the process or processing times changed after the start of manufacturing. Because the previous studies did not examine the changes of the state variables after the outset, they have not been suitable for supervising the internal processes when the batch is at work. On the other hand, the proposed methods can calculate the earliest/latest starting times in all processes, which contribute to keeping track of bottlenecks.

The primary focus was on calculating and predicting the internal states and on adjusting feeding times to follow the desired schedules. However, a part of repetitive MIMO-FIFO systems requires adjusting system parameters. On the other hand, this approach assumes that they are given, not adjustable. In considering an applications to transportation systems, the system parameters, which correspond to the required times between stations, could be handled as adjustable. This kind of system is currently outside the scope of this research, and an extension to cope with this issue remains as future work.

ACKNOWLEDGMENT

This research was partially supported by a research grant from the Mazda Foundation.

REFERENCES

Baccelli, F., Cohen, G., Olsder, G. J., and Quadrat, J. P. (1992), *Synchronization and Linearity, An Algebra for Discrete Event Systems*, John Wiley and Sons, New York.

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

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

Goto, H. and Masuda, S. (2004), Derivation algorithm of state-space equation based on max-plus algebra, Industrial Engineering and Management Systems, **3**, 1-11.

Goto, H. and Masuda, S. (2004), On the properties of the greatest subsolution for linear equations in the max-plus algebra, *IEICE Transactions on Fundamentals*, **E87-A**, 424-432.

Hillier, F. and Lieberman, G. (2002), *Introduction to Operations Research 7ed.*, McGraw-Hill Science, New York.

Ramamoorthy, C. and Ho, G. (1980), Performance evaluation of asynchronous concurrent systems using Petri nets, *IEEE Transactions on Software Engineering*, **SE-6**, 440-449.

Menguy, E., Boimond, J. L., Hardouin, L., and Ferrier, J. (2000), A first step towards adaptive control for linear systems in max algebra, *Discrete Event Dynamic Systems: Theory and Applications*, **10**, 347-367.

Masuda, S., Goto, H., Amemiya, T., and Takeyasu, K. (2003), An optimal inverse system for max-plus linear systems with linear parameter-varying structure, *Transactions of the Institute of System, Control and Information Engineers*, **16**, 44-53.

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