

Model predictive control-based scheduler for repetitive discrete event systems with capacity constraints

Hiroyuki Goto^a

^a Department of Industrial and System Engineering, Hosei University, Japan
Email: goto-h@hosei.ac.jp

(Received September 19, 2012; in final form April 04, 2013)

Abstract. A model predictive control-based scheduler for a class of discrete event systems is designed and developed. We focus on repetitive, multiple-input, multiple-output, and directed acyclic graph structured systems on which capacity constraints can be imposed. The target system's behaviour is described by linear equations in max-plus algebra, referred to as state-space representation. Assuming that the system's performance can be improved by paying additional cost, we adjust the system parameters and determine control inputs for which the reference output signals can be observed. The main contribution of this research is twofold, 1: For systems with capacity constraints, we derived an output prediction equation as functions of adjustable variables in a recursive form, 2: Regarding the construct for the system's representation, we improved the structure to accomplish general operations which are essential for adjusting the system parameters. The result of numerical simulation in a later section demonstrates the effectiveness of the developed controller.

Keywords: Model predictive control; max-plus algebra; capacity constraint; adjustable parameter; internal representation.

AMS Classification: 15A80, 37N35, 90C05, 93C65, 93C83

1. Introduction

The objective of this research is to design and implement a model predictive control (MPC) [1,2] based scheduler for a class of discrete event systems (DESs) [3,4]. A discrete event system is a dynamical system whose state evolves at discrete time instants.

We in particular focus on DESs with the following specifications: (1) repetitive, (2) multiple-input and multiple-output (MIMO), (3) first-in and first-out (FIFO), and (4) directed acyclic graph (DAG)-structured precedence relations [5,7]. In addition to these, we consider incurring capacity constraints between two arbitrary nodes.

The dynamics of such systems can be formulated by a pair of linear equations in max-plus algebra [8,9], and a system whose behaviour is described by these equations is referred to as a max-plus linear system. Typical examples

include manufacturing and transportation systems [10,11]. Since the form of the pair is similar to the state-space representation in modern control theory, several concepts in control theory have been applied to max-plus linear systems [12-15]. Amongst the related concepts, the frameworks for controlling manufacturing systems based on MPC provide cost effective schedulers, which can be roughly classified into two types.

One type deals with systems whose feature quantities of the system parameters are given [6], [16,18]. The control objective is to determine an optimal input time that minimises an objective function for given reference output times; the function includes an output error and a rate of change of input times. References [6], [16] consider a deterministic system, while [17] extends this to a stochastic case. Reference [18] takes into account external perturbation and considers stabilizing a system by feedback control. A typical feature of these works is that

the system parameters are either constant or vary stochastically around predetermined values.

The other type of framework handles systems in which the system parameters can be adjusted as well as the input times. For example, the framework in reference [7] first adjusts the system parameters to minimise both an operational cost and tardiness penalty, and then determines an input time which minimises a waiting cost. If we apply the framework to manufacturing systems, the controller aims to reduce the manufacturing cost, avoid delays for the due dates, and supply materials as late as possible.

Reference [19] proposed an efficient method to handle the system parameters in the output prediction equation as adjustable variables by introducing the concept of a 'cell'. The control framework is reduced to solving two optimisation problems in sequence. However, the application scope is currently restricted; the framework assumes that only a single job can be processed at one time in a single facility, and that the number of in-process jobs, which can exist between two adjacent facilities, is unlimited.

However, capacity constraints are occasionally incurred on several locations in most actual systems. Thus, a framework to control such systems needs to be developed. In view of this, we develop an MPC based scheduler in which maximum capacity constraints in a single or between two arbitrary facilities can be incurred. The main contribution of this research is twofold.

The first contribution is regarding the output prediction equation. Starting from the state-space representation for systems with capacity constraints, we derived a prediction equation as functions of adjustable variables. Since it is hard to represent the resulting matrices equation in an explicit form, each block of the matrices is expressed in a recursive form.

The second contribution is an improvement of the structure of the 'cell'. The previous structure is designed only for two special operations. However, in handling systems with capacity constraints, a framework for general addition and multiplication is required to construct an optimisation problem for adjusting the system parameters. We thus improve the structure and create procedures for these operations.

2. Preliminaries

2.1. Algebraic system

We first define a set $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$, where \mathbb{R} represents the real set. Then, for $x, y \in \mathbb{R}_{\max}$, we define three operators and two unit elements: $x \oplus y = \max(x, y)$, $x \otimes y = x + y$, $x \wedge y = \min(x, y)$, $\varepsilon (= -\infty)$, and $e (= 0)$. In a conventional formalism, the special rule $\varepsilon \otimes (-\varepsilon) = (-\varepsilon) \otimes \varepsilon = \varepsilon$ is adopted. For a number set Ω , if $x_l \in \mathbb{R}_{\max}$, $\bigoplus_{l \in \Omega} x_l = \{\max_{l \in \Omega} x_l : \text{if } \Omega \neq \emptyset, \varepsilon : \text{if } \Omega = \emptyset\}$ and $\bigwedge_{l \in \Omega} x_l = \{\min_{l \in \Omega} x_l : \text{if } \Omega \neq \emptyset, \varepsilon : \text{if } \Omega = \emptyset\}$. For vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}_{\max}^n$ and a scalar $a \in \mathbb{R}$, $\mathbf{x} \geq \mathbf{y}$ and $\mathbf{x} \geq a$ represents $[x]_i \geq [y]_i$ and $[x]_i \geq a$, respectively, for all i ($1 \leq i \leq n$). For matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}_{\max}^{m \times n}$, $\mathbf{Z} \in \mathbb{R}_{\max}^{n \times q}$, and $\mathbf{W} \in \mathbb{R}_{\max}^{m \times q}$, we define $[\mathbf{X} \oplus \mathbf{Y}]_{ij} = [\mathbf{X}]_{ij} \oplus [\mathbf{Y}]_{ij}$, $[\mathbf{X} \otimes \mathbf{Z}]_{ij} = \bigoplus_{l=1}^n ([\mathbf{X}]_{il} \otimes [\mathbf{Z}]_{lj})$, and $[\mathbf{X} \setminus \mathbf{W}]_{ij} = \bigwedge_{l=1}^n (-[\mathbf{X}]_{li} + [\mathbf{W}]_{lj})$. For the unit matrices, ε is a matrix, all elements of which are ε , while \mathbf{E} is a matrix whose diagonal elements are e and off-diagonal elements are ε . The priority of operators \otimes and \setminus is higher than that of \oplus and \wedge , and \otimes is often omitted when no confusion is likely to arise. If $\mathbf{X} \in \mathbb{R}_{\max}^{n \times n}$ and $b \in \mathbb{N}$, we define and write the b -th power of \mathbf{X} by $\mathbf{X}^{\otimes 0} = \mathbf{E}$ and $\mathbf{X}^{\otimes b} = \mathbf{X} \otimes \mathbf{X}^{\otimes (b-1)}$.

2.2. State-space representation

Let the token number, number of nodes, and number of external inputs and outputs be k, n, r , and m , respectively. We denote the processing, process start, process completion, input, and output times for the k -th token by $\mathbf{d}(k) \in \mathbb{R}_{\max}^n$, $\mathbf{x}^-(k) \in \mathbb{R}_{\max}^n$, $\mathbf{x}^+(k) \in \mathbb{R}_{\max}^n$, $\mathbf{u}(k) \in \mathbb{R}_{\max}^r$, and $\mathbf{y}(k) \in \mathbb{R}_{\max}^m$, respectively. In control theory terminology, $\mathbf{d}(k)$ and $\mathbf{u}(k)$ are referred to as the system parameter and control input, respectively, while $\mathbf{x}^-(k)$ and $\mathbf{x}^+(k)$ denote the state vector.

Moreover, we introduce the following representation matrices:

$$\mathbf{P}_k = \text{diag}[\mathbf{d}(k)],$$

$$\begin{aligned}
 [\mathbf{F}]_{ij} &= \begin{cases} f_{ij} : \text{transition time from node } j \text{ to} \\ \text{node } i, \\ \varepsilon : \text{if node } i \text{ is not a successor of} \\ \text{node } j, \end{cases} \\
 [\mathbf{H}^{(h)}]_{ij} &= \begin{cases} e : \text{maximum number of tokens} \\ \text{between node } i \text{ and its downstream} \\ \text{node } j \text{ is } h. \\ \varepsilon : \text{if otherwise,} \end{cases} \\
 [\mathbf{B}]_{ij} &= \begin{cases} b_{ij} : \text{transition time from external} \\ \text{input } j \text{ to node } i, \\ \varepsilon : \text{if node } i \text{ does not have an} \\ \text{external input } j, \end{cases} \\
 [\mathbf{C}]_{ij} &= \begin{cases} c_{ij} : \text{transition time from node } j \text{ to} \\ \text{external output } i, \\ \varepsilon : \text{if node } j \text{ does not have an} \\ \text{external output } i, \end{cases}
 \end{aligned}$$

where $\mathbf{P}_k \in \mathbb{R}_{\max}^{n \times n}$, $\mathbf{F} \in \mathbb{R}_{\max}^{n \times n}$, $\mathbf{H}^{(h)} \in \mathbb{R}_{\max}^{n \times n}$, $\mathbf{B} \in \mathbb{R}_{\max}^{n \times r}$, and $\mathbf{C} \in \mathbb{R}_{\max}^{m \times n}$ are referred to as the weight, adjacency, capacity, input, and output matrices, respectively.

Then, the earliest process start, process completion, and output times of the k -th token are represented by the following equations [20]:

$$\mathbf{x}(k) = \bar{\mathbf{A}}_k [\bigoplus_{h=1}^Q \bar{\mathbf{H}}^{(h)} \mathbf{x}(k-h) \oplus \bar{\mathbf{B}} \mathbf{u}(k)], \quad (1)$$

$$\mathbf{y}(k) = \bar{\mathbf{C}} \mathbf{x}(k), \quad (2)$$

where

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}^-(k) \\ \mathbf{x}^+(k) \end{bmatrix}, \quad \bar{\mathbf{A}}_k = \begin{bmatrix} (\mathbf{F}\mathbf{P}_k)^* & (\mathbf{F}\mathbf{P}_k)^* \mathbf{F} \\ (\mathbf{P}_k \mathbf{F})^* \mathbf{P}_k & (\mathbf{P}_k \mathbf{F})^* \end{bmatrix},$$

$$\bar{\mathbf{H}}^{(1)} = \begin{bmatrix} \mathbf{E} & \mathbf{H}^{(1)} \\ \varepsilon & \mathbf{E} \end{bmatrix}, \quad \bar{\mathbf{H}}^{(h)} = \begin{bmatrix} \varepsilon & \mathbf{H}^{(h)} \\ \varepsilon & \varepsilon \end{bmatrix} \quad h \geq 2, \quad (3)$$

$$\bar{\mathbf{B}} = [\mathbf{B}^T \ \varepsilon^T]^T, \quad \bar{\mathbf{C}} = [\varepsilon \ \mathbf{C}].$$

Equations (1) and (2) are known as the state and output equations, respectively. Q is the maximum capacity of the system. Operator $*$ is called the Kleene star. For a DAG structured adjacency matrix $\mathbf{G} \in \mathbb{R}_{\max}^{n \times n}$, the Kleene star of \mathbf{G} is calculated as

$$\mathbf{G}^* = \bigoplus_{l=0}^{s-1} \mathbf{G}^{\otimes l} = \mathbf{E} \oplus \mathbf{G} \oplus \dots \oplus \mathbf{G}^{\otimes(s-1)},$$

where s is an instance that satisfies $\mathbf{G}^{\otimes(s-1)} \neq \varepsilon$ and $\mathbf{G}^{\otimes s} = \varepsilon$ ($1 \leq s \leq n$).

2.3. Design of an MPC controller

The design of an MPC controller generally consists of the following three steps.

- 1) Derive prediction and output equations
- 2) Determine control inputs
- 3) Apply the receding horizon method

In the first step, we apply the state equation Eq. (1) iteratively and derive prediction equations for the $(k, k+1, \dots, k+N-1)$ -th ($N \geq 1$) tokens, where N is known as the prediction horizon. We then apply the output equation Eq. (2) to the prediction equations, for which the output times of the corresponding tokens are predicted.

In the second step, we set reference signals for the output equations, which correspond to give target output times. We then construct an optimisation problem to determine a control law. To reduce the computation load, a technique to decrease the number of variables is often used. Given a certain constant $N_c (\leq N)$, we assume that the adjustable variables for the $(k+N_c-1, k+N_c, \dots, k+N-1)$ -th tokens remain unchanged, where N_c is known as control horizon. The variables include the control input $\mathbf{u}(k)$ and system parameter $\mathbf{d}(k)$.

In respect to the optimal parameters for the $(k, k+1, \dots, k+N-1)$ -th tokens, we apply only the first one, and the succeeding laws are determined after the token number is incremented and the parameters recalculated. This method is known as the receding horizon method, by which a feedback control is achieved.

3. Proposed Framework

In developing an MPC framework for systems with capacity constraints, first, we derive the prediction and output equations. Then, the system parameters are adjusted by solving an optimisation problem, and subsequently the optimal input time is determined.

3.1. Prediction equation

The prediction equation can be derived using Eq. (1). Letting the prediction horizon be N , we formally express the prediction equation for the $(k+l)$ -th ($0 \leq l \leq N-1$) token as follows:

$$\mathbf{x}(k+l) = \bigoplus_{h=1}^Q \Phi_{l,h} \mathbf{x}(k-h) \oplus \bigoplus_{p=0}^l \mathbf{A}_{l,p} \mathbf{u}(k+p) \quad (4)$$

For $l=0$, by comparing the coefficients of $\mathbf{x}(k-h)$ ($1 \leq h \leq Q$) and $\mathbf{u}(k)$ in Eqs. (1) and (4), we have

$$\Phi_{0,h} = \bar{A}_k \bar{H}^{(h)} \quad (1 \leq h \leq Q), \quad \Delta_{0,0} = \bar{A}_k \bar{B}. \quad (5)$$

If $l \geq 1$, we advance the token number k in Eq. (1) for l steps, by which

$$\mathbf{x}(k+l) = \bar{A}_{k+l} \otimes \left[\bigoplus_{m=1}^Q \bar{H}^{(m)} \mathbf{x}(k+l-m) \oplus \bar{B} \mathbf{u}(k+l) \right]. \quad (6)$$

is obtained. On the other hand, by replacing l with $l-m$ in Eq. (4), we obtain

$$\mathbf{x}(k+l-m) = \bigoplus_{h=1}^Q \Phi_{l-m,h} \mathbf{x}(k-h) \oplus \bigoplus_{p=0}^{l-m} \Delta_{l-m,p} \mathbf{u}(k+p). \quad (7)$$

If $l \geq Q$, we substitute this for the first term on the right-hand side in Eq. (6). This yields

$$\begin{aligned} \mathbf{x}(k+l) &= \bar{A}_{k+l} \otimes \\ & \left[\bigoplus_{h=1}^Q \bigoplus_{m=1}^Q \bar{H}^{(m)} \Phi_{l-m,h} \mathbf{x}(k-h) \right. \\ & \left. \oplus \bigoplus_{m=1}^Q \bigoplus_{p=0}^{l-m} \bar{H}^{(m)} \Delta_{l-m,p} \mathbf{u}(k+p) \right. \\ & \left. \oplus \bar{B} \mathbf{u}(k+l) \right]. \quad (8) \end{aligned}$$

The comparison of the coefficient for $\mathbf{x}(k-h)$ in Eqs. (4) and (8) yields

$$\Phi_{l,h} = \bigoplus_{m=1}^Q \bar{A}_{k+l} \bar{H}^{(m)} \Phi_{l-m,h}. \quad (9)$$

On the other hand, if $l < Q$, Eq. (6) can be expanded as

$$\mathbf{x}(k+l) = \bar{A}_{k+l} \left[\bigoplus_{m=1}^l \bar{H}^{(m)} \mathbf{x}(k+l-m) \oplus \bigoplus_{m=l+1}^Q \bar{H}^{(m)} \mathbf{x}(k+l-m) \oplus \bar{B} \mathbf{u}(k+l) \right]. \quad (10)$$

The second term on the right-hand side can be transformed into

$$\begin{aligned} \bigoplus_{m=l+1}^Q \bar{H}^{(m)} \mathbf{x}(k+l-m) \\ = \bigoplus_{h=1}^{Q-l} \bar{H}^{(l+h)} \mathbf{x}(k-h) \end{aligned} \quad (11)$$

Using Eqs. (7) and (11), Eq. (10) is represented as follows:

$$\begin{aligned} \mathbf{x}(k+l) &= \bar{A}_{k+l} \otimes \\ & \left[\bigoplus_{h=1}^Q \bigoplus_{m=1}^l \bar{H}^{(m)} \Phi_{l-m,h} \mathbf{x}(k-h) \right. \\ & \left. \oplus \bigoplus_{m=1}^l \bigoplus_{p=0}^{l-m} \bar{H}^{(m)} \Delta_{l-m,p} \mathbf{u}(k+p) \right. \\ & \left. \oplus \bigoplus_{h=1}^{Q-l} \bar{H}^{(l+h)} \mathbf{x}(k-h) \oplus \bar{B} \mathbf{u}(k+l) \right] \end{aligned} \quad (12)$$

Let us compare the coefficients of $\mathbf{x}(k-h)$ in Eqs. (4) and (12) again. If $h \leq Q-l$, i.e., $l \leq Q-h$, we obtain the following relationship

$$\Phi_{l,h} = \bigoplus_{m=1}^l \bar{A}_{k+l} \bar{H}^{(m)} \Phi_{l-m,h} \oplus \bar{A}_{k+l} \bar{H}^{(l+h)}. \quad (13)$$

On the other hand, if $h > Q-l$, i.e., $l > Q-h$, the third term on the right-hand side in Eq. (12) can be omitted and we obtain

$$\Phi_{l,h} = \bigoplus_{m=1}^l \bar{A}_{k+l} \bar{H}^{(m)} \Phi_{l-m,h}. \quad (14)$$

In view of Eqs. (9) and (14), $\Phi_{l,h}$ can be expressed in the following unified manner

$$\Phi_{l,h} = \bigoplus_{m=1}^{Q \wedge l} \bar{A}_{k+l} \bar{H}^{(m)} \Phi_{l-m,h}, \quad (15)$$

for $l > Q-h$.

Next, we obtain $\Delta_{l,p}$ by comparing the coefficients of $\mathbf{u}(k+p)$ in Eqs. (4), (8), and (12). If $p=l$, only the last terms in these equations are effective and we obtain

$$\Delta_{l,l} = \bar{A}_{k+l} \bar{B}.$$

If $p < l$ and $l \geq Q$, the comparison of $\mathbf{u}(k+p)$ in Eqs. (4) and (8) yields

$$\Delta_{l,p} = \bigoplus_{m=1}^Q \bar{A}_{k+l} \bar{H}^{(m)} \Delta_{l-m,p}. \quad (16)$$

If $p < l$ and $l < Q$, the comparison of $\mathbf{u}(k+p)$ in Eqs. (4) and (12) leads to

$$\Delta_{l,p} = \bigoplus_{m=1}^l \bar{A}_{k+l} \bar{H}^{(m)} \Delta_{l-m,p}. \quad (17)$$

Equations (16) and (17) can be then unified to

$$\Delta_{l,p} = \bigoplus_{m=1}^{Q \wedge l} \bar{A}_{k+l} \bar{H}^{(m)} \Delta_{l-m,p}, \quad (18)$$

for $p < l$.

Furthermore, we derive an output prediction equation for the $(k, k+1, \dots, k+N-1)$ -th tokens. Using Eqs. (2) and (4), the simultaneous output prediction equations for these tokens can be expressed as

$$\mathbf{Y}(k) = \bar{\Phi} \mathbf{X}(k) \oplus \bar{\mathbf{A}} \mathbf{U}(k), \quad (19)$$

where

$$\begin{aligned} \mathbf{Y}(k) &= [\mathbf{y}^T(k) \mathbf{y}^T(k+1) \cdots \mathbf{y}^T(k+N-1)]^T, \\ \mathbf{X}(k) &= [\mathbf{x}^T(k-1) \mathbf{x}^T(k-2) \cdots \mathbf{x}^T(k-Q)]^T, \\ \mathbf{U}(k) &= [\mathbf{u}^T(k) \mathbf{u}^T(k+1) \cdots \mathbf{u}^T(k+N-1)]^T, \end{aligned}$$

$$\bar{\Phi} = \begin{bmatrix} \bar{C}\Phi_{0,1} & \bar{C}\Phi_{0,2} & \cdots & \bar{C}\Phi_{0,Q} \\ \bar{C}\Phi_{1,1} & \bar{C}\Phi_{1,2} & \ddots & \bar{C}\Phi_{1,Q} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{C}\Phi_{N-1,1} & \bar{C}\Phi_{N-1,2} & \cdots & \bar{C}\Phi_{N-1,Q} \end{bmatrix}, \quad (20)$$

$$\bar{A} = \begin{bmatrix} \bar{C}A_{0,0} & \varepsilon & \cdots & \varepsilon \\ \bar{C}A_{1,0} & \bar{C}A_{1,1} & & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ \bar{C}A_{N-1,0} & \bar{C}A_{N-1,1} & \cdots & \bar{C}A_{N-1,N-1} \end{bmatrix}.$$

In this manner, the coefficient matrices $\bar{\Phi}$ and \bar{A} in the output prediction equation (19) are now derived. These matrices are functions of the system parameters $\mathbf{d}(k+l)$ ($0 \leq l \leq N-1$) handled as adjustable variables, each block of which is expressed in a recursive form.

3.2. Control objective

The set of output prediction equations expressed in Eq. (19) is used to construct a control law for which the given reference output times are observed.

Let the reference output times for the $(k, k+1, \dots, k+N-1)$ -th tokens be $\mathbf{r}(k)$, $\mathbf{r}(k+1)$, \dots , $\mathbf{r}(k+N-1)$, respectively. The well-known control laws are to adjust $\mathbf{d}(k+l)$ and/or $\mathbf{u}(k+l)$ to satisfy

$$\bar{\Phi}\mathbf{X}(k) \oplus \bar{A}\mathbf{U}(k) \leq \mathbf{R}(k), \quad (21)$$

where

$$\mathbf{R}(k) = [\mathbf{r}^T(k) \mathbf{r}^T(k+1) \cdots \mathbf{r}^T(k+N-1)]^T.$$

We assume that the operational and waiting costs can be reduced if we can enlarge $\mathbf{d}(k+l)$ and $\mathbf{u}(k+l)$, respectively. On the other hand, a tardiness penalty is incurred if an actual output time exceeds the reference time. A control law is then designed by taking into account these three points.

Let $\boldsymbol{\mu}_l$ be the tardiness vector for the $(k+l)$ -th ($0 \leq l \leq N-1$) token, i.e.

$$[\boldsymbol{\mu}_l]_i = \max\{[\mathbf{y}(k+l)]_i - [\mathbf{r}(k+l)]_i, 0\} \quad (1 \leq i \leq m).$$

One natural idea for control objective is, recalling Eq. (21), to enlarge $\mathbf{d}(k+l)$ and $\mathbf{u}(k+l)$, and decrease $\boldsymbol{\mu}_l$ under a constraint

$$\bar{\Phi}\mathbf{X}(k) \oplus \bar{A}\mathbf{U}(k) \leq \mathbf{R}(k) + \bar{\boldsymbol{\mu}}, \quad (22)$$

where $\bar{\boldsymbol{\mu}} = [\boldsymbol{\mu}_0^T \boldsymbol{\mu}_1^T \cdots \boldsymbol{\mu}_{N-1}^T]^T \in \mathbb{R}^{m \cdot N}$ can be interpreted as the set of artificial variables.

However, if we formulate this objective in a straightforward manner, the number of variables and constraints increases very sharply as the prediction horizon increases, and the computation time inflates accordingly. Hence, we use a lightweight method used in [21] which is understood as a greedy method; we determine $\mathbf{d}(k+l)$ and $\mathbf{u}(k+l)$ in two separate steps.

3.3. Adjustment of the system parameters

An optimisation problem is constructed and solved to adjust the system parameters $\mathbf{d}(k+l)$ ($0 \leq l \leq N-1$). First, we ignore the effect of the input times $\mathbf{U}(k)$ in Eq. (22), and formulate a control objective as

$$\bar{\Phi}\mathbf{X}(k) \leq \mathbf{R}(k) + \bar{\boldsymbol{\mu}}, \quad (23)$$

where $\bar{\boldsymbol{\mu}} = [\boldsymbol{\mu}_0^T \boldsymbol{\mu}_1^T \cdots \boldsymbol{\mu}_{N-1}^T]^T \in \mathbb{R}^{m \cdot N}$ and

$$\boldsymbol{\mu}_l \geq \mathbf{0} \text{ for all } l \text{ (} 0 \leq l \leq N-1 \text{)}. \quad (24)$$

Recalling Eq. (20), the j -th row of the l -th row block in Eq. (23) ($1 \leq j \leq m, 0 \leq l \leq N-1$) can be rewritten as

$$\bigoplus_{h=1}^Q \bigoplus_{i=1}^n [\mathbf{C}\Phi_{l,h}]_{ji} \otimes [\mathbf{x}(k-h)]_i \leq [\mathbf{r}(k+l)]_j + [\boldsymbol{\mu}_l]_j,$$

for all j and l ($1 \leq j \leq m, 0 \leq l \leq N-1$). This can be further reduced to the following simultaneous inequalities.

$$[\mathbf{C}\Phi_{l,h}]_{ji} - [\boldsymbol{\mu}_l]_j \leq [\mathbf{r}(k+l)]_j - [\mathbf{x}(k-h)]_i, \quad (25)$$

for all h, i, j , and l ($1 \leq h \leq Q, 1 \leq i \leq n, 1 \leq j \leq m, 0 \leq l \leq N-1$). For the other constraints, we assume that the lower and upper bounds on the system parameters are as follows:

$$0 \leq \mathbf{d}_{\min} \leq \mathbf{d}(k+l) \leq \mathbf{d}_{\max} \quad (0 \leq l \leq N-1). \quad (26)$$

With respect to the objective function, we consider reducing both the operational cost and tardiness, and formulate the objective function as

$$J = - \sum_{l=0}^{N-1} \sum_{i=1}^n \alpha_i \cdot [\mathbf{d}(k+l)]_i + \sum_{l=0}^{N-1} \sum_{j=1}^m \beta_j \cdot [\boldsymbol{\mu}_l]_j, \quad (27)$$

where α_i and β_j (≥ 0) are the penalties for operational cost and tardiness, respectively.

Consequently, the optimisation problem for $\mathbf{d}(k+l)$ and $\boldsymbol{\mu}_l$ is summarised as

$$J \rightarrow \min \text{ subject to Eqs. (24)–(26)}. \quad (28)$$

Since this problem has the form of a linear programming problem, we can solve the problem by using an efficient algorithm such as the interior point method [22].

In addition to these, we consider introducing the concept of control horizon to reduce the number of variables. Letting the control horizon be N_c , we assume

$$[\mathbf{d}(k+l)]_i = [\mathbf{d}(k+N_c-1)]_i \text{ for all } l \\ (N_c - 1 \leq l \leq N - 1).$$

Applying this concept, the number of system parameters can be decreased from $N \cdot n$ to $N_c \cdot n$. According to [21], we can obtain sufficient performance even if we set $N_c = 1$.

3.4. Determination of the input times

In the second step of the control law, the input times $\mathbf{U}(k)$ are determined. If the control inputs are small enough not to affect the output times, the earliest output times are predicted as

$$\mathbf{Y}(k) = \overline{\Phi} \mathbf{X}(k).$$

If the values of the state vector are sufficiently large, $\mathbf{Y}(k)$ may exceed the reference output time $\mathbf{R}(k)$. For such cases, we aim to adjust the input times to complete the k -th token at $\mathbf{Y}(k)$ and not at $\mathbf{R}(k)$. This policy can be expressed as

$$\overline{\Delta} \mathbf{U}(k) \leq \overline{\Phi} \mathbf{X}(k) \oplus \mathbf{R}(k). \quad (29)$$

We should note here that the term $\overline{\Phi} \mathbf{X}(k)$ appears on the right-hand side compared to Eq. (21). In Eq. (29), only $\mathbf{U}(k)$ is a variable vector, while the others are constant vectors and matrices.

To reduce the waiting cost, the input times should be as large as possible in observance of Eq. (29). Using a lightweight formula known as the residuation theory [5], the optimal solution for $\mathbf{U}(k)$ is calculated as follows:

$$\mathbf{U}(k) = \overline{\Delta} \setminus [\overline{\Phi} \mathbf{X}(k) \oplus \mathbf{R}(k)]. \quad (30)$$

The optimal input times $\mathbf{u}(k)$, $\mathbf{u}(k+1)$, \dots , $\mathbf{u}(k+N-1)$ are then obtained using this result.

In the last step, we consider applying the concept of receding horizon method to achieve a feedback control. For the obtained optimal input times, we give an actual control input only for the first step. The inputs for the subsequent steps are recalculated after k is incremented and the values of the latest state vector are obtained. We adopt and give only the first block of Eq. (30) for

the optimal input, the specific form of which is expressed as

$$\mathbf{u}(k) = \bigwedge_{l=0}^{N-1} \{(\overline{\mathbf{C}} \mathbf{A}_{l,0}) \setminus \mathbf{z}(k+l)\}, \quad (31)$$

where

$$\mathbf{z}(k+l) = \bigoplus_{h=1}^Q \overline{\mathbf{C}} \Phi_{l,h} \mathbf{x}(k-h) \oplus \mathbf{r}(k+l).$$

Using the optimal solutions given in Eqs. (28) and (31), we can operate the system at a lower operational cost observing the reference output times.

3.5. Improvement of the cell's structure

To solve the optimisation problem constructed in section 3.3, a method for deriving and representing $\Phi_{l,h}$ in Eq. (25) as a function of the system parameters $\mathbf{d}(k+l)$ ($0 \leq l \leq N-1$) is essential. This matrix is calculated iteratively using Eqs. (5), (13), (15), and (18), and $\overline{\mathbf{A}}_k$ includes the Kleene star of the weighted adjacency matrices as shown in Eq. (3).

Reference [19] proposed a method for computing the transition matrix in the state-space representation as a function of the system parameters by introducing the concept of a 'cell'. However, the method is designed only for calculating the Kleene star and multiplying a diagonal matrix.

On the other hand, in handling systems with capacity constraints, a framework for general addition and multiplication in max-plus algebra is required. Thus, we improve the structure of a cell.

Figure 1 depicts the improved structure and an example. Each resulting matrix is described by collection of cells and each element of which is described by one cell. For a given matrix \mathbf{X} , Src and Dst store the column and row numbers of matrix \mathbf{X} , respectively. If there is no cell that has (Src, Dst) = (j, i) , this indicates $[\mathbf{X}]_{ij} = \varepsilon$. Arrays Coef() and Pows() are used for representing the coefficient and power of a function of the system parameters $\mathbf{d}(k+l)$ ($0 \leq l \leq N_c - 1$). Arrays Mins() and Maxs() are used for recording the minimum and maximum weighted sums for given lower and upper bounds of the system parameters \mathbf{d}_{\min} and \mathbf{d}_{\max} .

The arrays' size is equal to the number of irreducible terms. Let S and s be the number of irreducible terms and the term number in the cell, respectively. Then, the type and size of the arrays are Coef() $\in \mathbb{R}^S$, Pows() $\in \mathbb{Z}^{S \times (n \cdot N_c)}$, Mins() $\in \mathbb{R}^S$ and Maxs() $\in \mathbb{R}^S$.

We now consider the case for $\mathbf{X} \in \mathbb{F}_{\max}^{3 \times 3}$, $\mathbf{d}_{\min} = (1, 2, 4)$, $\mathbf{d}_{\max} = (3, 3, 8)$, and $N = N_c = 2$. Then, the cell in Figure 1 indicates $[\mathbf{X}]_{31} = 3 \otimes d_1(k) \otimes d_3(k+1) \oplus 1 \otimes d_2(k)^2$, $S = 2$, $[\mathbf{d}_{\min}]_1 \otimes [\mathbf{d}_{\min}]_3 = 1 \otimes 4 = 5$, $[\mathbf{d}_{\max}]_1 \otimes [\mathbf{d}_{\max}]_3 = 3 \otimes 8 = 11$, $[\mathbf{d}_{\min}]_2^{\otimes 2} = 4$, and $[\mathbf{d}_{\max}]_2^{\otimes 2} = 6$.

Using this structure, the resulting matrices in the state-space representation and output equations can be represented as functions of the system parameters. Since the computations for the Kleene star of a matrix and the multiplication of a diagonal matrix have already been developed in [19], we newly develop the remaining operations required to derive the output prediction equation.

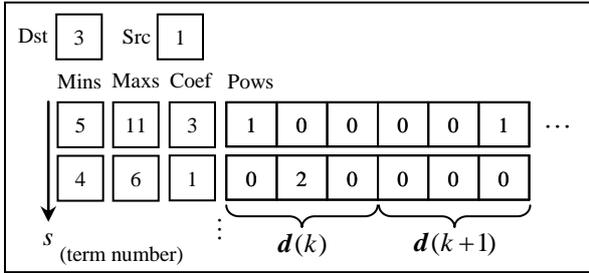


Figure 1. The structure and an example of a cell.

3.5.1. Addition

Two given cells with equal Src and Dst can simply be added by concatenating the Coef(), Pows(), Mins(), and Maxs() of the two cells. However, a naïve concatenation of these may yield redundant terms. Thus, a procedure for removing the redundant terms, which is considered as a branch and bound [23], is needed. This procedure is repeated for all rows and columns of the given matrices.

3.5.2. Multiplication

To compute $\mathbf{X} \otimes \mathbf{Z}$ for matrices $\mathbf{X} \in \mathbb{R}_{\max}^{m \times n}$ and $\mathbf{Z} \in \mathbb{R}_{\max}^{n \times q}$, the calculation of $[\mathbf{X}]_{il} \otimes [\mathbf{Z}]_{lj}$ for all l ($1 \leq l \leq n$) is essential. For two given cells x and z that have $(\text{Src}, \text{Dst}) = (l, i)$ and $(\text{Src}, \text{Dst}) = (j, l)$, the resulting cell has $(\text{Src}, \text{Dst}) = (j, i)$. Let the number of terms in cells x and z and their term numbers be S_x , S_z , s_x , and s_z , respectively. Then, recalling the property of the distributive law, arrays Coef(), Pows(), Maxs(), and Mins() are multiplied (added in conventional algebra) for all pairs of s_x and s_z

($1 \leq s_x \leq S_x, 1 \leq s_z \leq S_z$). If cell x does not exist, i.e., $s_x = 0$, the resulting cell is equal to cell z and (Src, Dst) is changed to (j, i) , and vice versa. The addition of the resulting cell to the previous result is repeated for all l ($1 \leq l \leq n$).

3.5.3. Branch and bound

After the above addition or multiplication above, the resulting cell may have redundant terms that have no effect on the result. Such terms should be removed from the cell to reduce the computation time. For a target cell x , we can remove the terms that cannot be the maximum value. In the first step, we remove terms s ($1 \leq s \leq S, s \neq s_M$) that satisfy

$$\text{Maxs}(s) \otimes \text{Coef}(s) \leq \text{Mins}(s_M) \otimes \text{Coef}(s_M)$$

where

$$s_M = \text{argmax}_{j=1,2,\dots,S} [\text{Mins}(j) \otimes \text{Coef}(j)].$$

In the case of Figure 1, $s_M = 1$ holds and the second term $s = 2$ is removed.

Next, we evaluate the order relations for all pairs of terms i and j ($1 \leq i, j \leq S, i \neq j$). If

$$\text{Coef}(i) \leq \text{Coef}(j) \text{ and } \text{Pows}(i, l) \leq \text{Pows}(j, l)$$

for all l ($1 \leq l \leq n \cdot N$), the value of term i is constantly equal to or smaller than term j , and thus term i can be removed from the current cell. If tight bounds were set for $\mathbf{d}(k+1)$ in Eq. (26), the significant number of resulting terms would be reduced. In view of this, this procedure can be seen as a kind of branch and bound.

3.5.4. Difference from the previous structure

The difference between the previous structure in [19] and the proposed one in this paper is that element Dst, and arrays Mins() and Coef() are appended. By appending Dst, the structure of a cell becomes self-descriptive and general operation for multiplication is achieved. Array Mins() is used in the branch and bound procedure, and plays a role in reducing the complexity of the resulting matrix. The original objective of appending array Coef() is to improve the descriptive ability, and the direct benefit here is that a lag time between two successive processes can be taken into account.

4. Numerical Simulation

We present an application example of a manufacturing system. Figure 2 depicts a simple tandem structured manufacturing system with four facilities, one external input, and one

external output. The values in parentheses (*) represent the system parameters, which correspond to the processing times, while the values in brackets [*] represent the maximum number of in-process jobs that can exist in a single facility or between two facilities. For instance, facility 2 can process a maximum of three jobs, and the maximum number of in-process jobs that can exist between facilities 2 and 3 is five. Since the ratio of the maximum capacity [*] to the system parameter (*) gives the effective throughput (jobs per unit time) for each facility, facility 4 provides the highest throughput. We assume that the transition times between two adjacent facilities, starting upstream, are 0.2, 0.3, and 0.4, respectively. By incurring an additional operational cost, the processing times can be reduced, the minimum values of which are 0.5 less than the original ones. Then, we have the following representation matrices and vectors:

$$F = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 0.2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 0.3 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 0.4 & \varepsilon \end{bmatrix}, Q = \begin{bmatrix} 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 3 & 5 & \varepsilon \\ \varepsilon & \varepsilon & 1 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 2 \end{bmatrix},$$

$$B = \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix}, C = \begin{bmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \\ e \end{bmatrix}^T, d_{\min} = \begin{bmatrix} 1.5 \\ 4.5 \\ 1.5 \\ 2.5 \end{bmatrix}, d_{\max} = \begin{bmatrix} 2 \\ 5 \\ 2 \\ 3 \end{bmatrix}.$$

The capacity matrices $H^{(h)}$ are generated by executing the following computation for all h ($1 \leq h \leq 5$):

$$[H^{(h)}]_{ij} = \begin{cases} e & \text{if } [Q]_{ij} = h, \\ \varepsilon & \text{otherwise.} \end{cases}$$

For the weights in the objective function Eq. (27), we assume:

$$(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (1, 3, 1, 2), \beta_1 = 100.$$

Now we simulate the schedule for $40 (\equiv K)$ jobs. For reference output times, we set $r(1) = 0$, $r(k) = r(k-1) + 2.0$ for $(2 \leq k \leq 5, 11 \leq k \leq 20, 26 \leq k \leq 40)$, and $r(k) = r(k-1) + 1.0$ for $(6 \leq k \leq 10, 21 \leq k \leq 25)$. This means that the output interval in the normal state is 2.0, but this is reduced by half for $6 \leq k \leq 10$ and $21 \leq k \leq 25$. Thus, the facilities must speed up in advance for these perturbations.

Figure 3 illustrates the profile of the system parameter $d_2(k)$ for varying prediction horizons $N=1, 2, 5, 10,$ and 15 with control horizon $N_c=1$. At the beginning stage, the parameter is set to the normal value ($=5.0$) for all cases. Then, for $N=1$, it decreases sharply to the minimum value 4.5 at $k=9$ and 23. This phenomenon occurs 2 or 3 steps after the interval of the reference output time has been decreased. This lag is due to the capacity of facility 3. For $N=2$, the parameter is decreased at an earlier stage than for $N=1$. For $N=5$ and 10, the parameter decreases and recovers more smoothly at even an earlier stage, and does not decrease to the minimum limit. Furthermore, for $N=15$, the parameter does not decrease and remains constant all through the jobs.

Figure 4 depicts the output error for $N=1, 2, 5, 10,$ and 15 with $N_c=1$. We define the error for job k as $\max\{[y(k)]_j - [r(k)]_j, 0\}$, which can be interpreted as tardiness for the reference output time. For $N=1$ and 2, tardiness occurs soon after the interval of the reference output time has been decreased. By contrast, for $N=5$, the output time is moved up before $k=5$ and 20 to avoid tardiness, but tardiness eventually occurs at $k=10$ and 25. For $N=10$, the output times are moved up at $1 \leq k \leq 9$ and $12 \leq k \leq 24$, and tardiness is completely avoided. Tardiness is also avoided for $N=15$ and the output times are further moved up. In the light of Figures 3 and 4, if we set the prediction horizon N to a large value, we can avoid tardiness for the reference output time with only a small additional operational cost.

Table 1 shows the number of constraints for the linear programming problem in Eq. (28) and the performance of the controller. We define the performance index T as

$$T = -\sum_{k=1}^K \sum_{i=1}^n \alpha_i [d(k)]_i + \sum_{k=1}^K \sum_{j=1}^m \beta_j \max\{[y(k)]_j - [r(k)]_j, 0\}$$

where the first and second terms represent the sum of the operational cost and penalty for tardiness, respectively. Clearly, the performance is improved as the prediction horizon N increases. On the other hand, since the number of constraints increases, so too does the computation time to obtain the internal representation of the output prediction equations and to solve the linear programming problem.

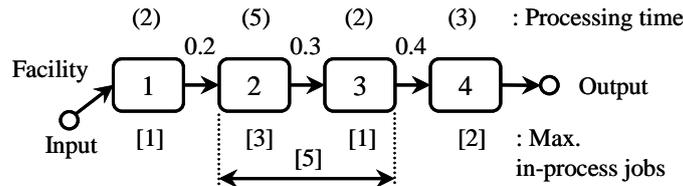


Figure 2. A simple system with four facilities

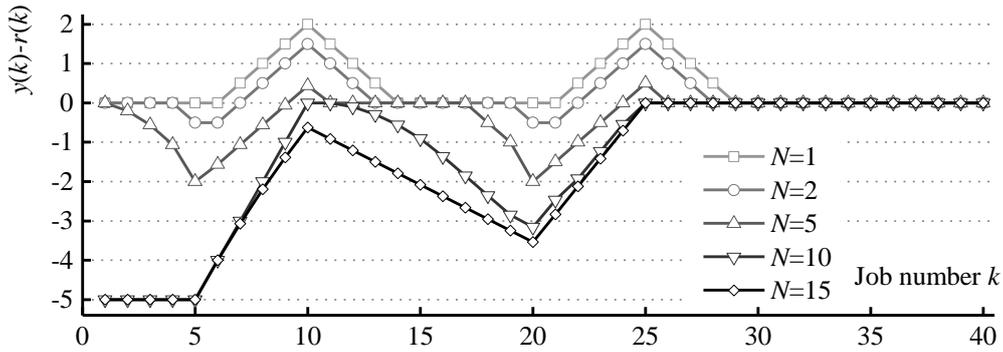


Figure 3. System parameter $d_2(k)$ for varying $N=1, 2, 5,$ and $10.$

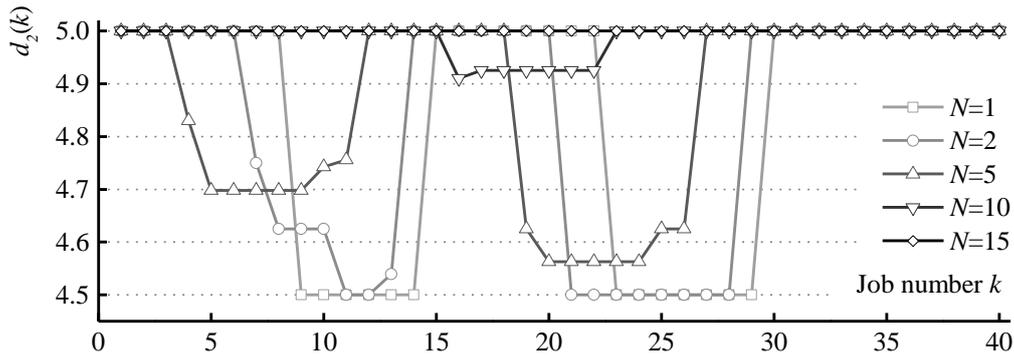


Figure 4. Output error for varying $N=1, 2, 5,$ and $10.$

Table 1. The number of constraint inequalities.

Prediction horizon	$N=1$	$N=2$	$N=5$	$N=10$	$N=15$
Num. constraints	11	26	145	934	3,533
Performance T	666.50	-34.07	-845.45	-980.09	-982.79

5. Conclusion

We have developed an MPC controller for a class of MIMO-FIFO discrete event systems with adjustable system parameters, in which capacity constraints exist in a single facility or between two arbitrary facilities. We first derived an output prediction equation as functions of the system parameters, each block of the resulting matrices of which is obtained in a recursive form.

We then improved the structure of a cell for system’s internal representation, and created procedures for general addition and multiplication in max-plus algebra. Through a numerical simulation, we have demonstrated that the proposed MPC controller achieves good performance if the prediction horizon is set to a large value.

Acknowledgments

This research has been supported in part by the research grant from JSPS Grants-in-Aid for Scientific Research No. 23510163.

References

- [1] Garcia, C.E., Prett, D.M., and Morari, M., Model predictive control: theory and practice – a Survey. *Automatica*, 25 (3), 335–348 (1989).
- [2] Wang, L., *Model Predictive Control System Design and Implementation Using MATLAB®*. Springer, New York (2009).
- [3] Cassandras, C.G. and Lafortune, S., *Introduction to Discrete Event Systems*. Springer, New York (2010).
- [4] Hrz, B. and Zhou, M., *Modeling and Control of Discrete-event Dynamic Systems: with Petri Nets and Other Tools*. Springer, London (2007).
- [5] Amari, S., Demongodin, I., Loiseau, J.J., and Martinez, C., Max-plus control design for temporal constraints meeting in timed event graphs. *IEEE Transactions on Automatic Control*, 57 (2), 462–467, 2011.
- [6] De Schutter, B., and Van den Boom, T.J.J., Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37 (7), 1049–1056 (2001).
- [7] Goto, H., Masuda, S., Takeyasu, K., and Amemiya, T., Derivation algorithm of state-space equation for production systems based on max-plus algebra. *Industrial Engineering and Management Systems*, 3 (1), 1–11 (2004).
- [8] Baccelli, F., Cohen, G., Olsder, G.J., and Quadrat, J.P., *Synchronization and Linearity. An Algebra for Discrete Event Systems*. Wiley, New York (1992) Available from: <http://maxplus.org>
- [9] Heidergott, B., Olsder, G.J., and Woude, L., *Max Plus at Work. Modeling and Analysis of Synchronized Systems*. Princeton University Press, New Jersey (2006).
- [10] Imaev, A. and Judd, R.P., Block diagram-based modeling of manufacturing systems using max-plus algebra. *Proceedings of the American Control Conference*, 4711–4716 (2009).
- [11] Goverde, R.M.P, A delay propagation algorithm for large-scale railway traffic networks. *Transportation Research Part C*, 18 (3), 269–287 (2010).
- [12] Akian, M., Gaubert, S., and Lakhoua, A., Convergence analysis of the max-plus finite element method for solving deterministic optimal control problems. *Proceedings of the 47th IEEE Conference on Decision and Control*, 927–934 (2008).
- [13] Maia, C.A., Andrade, C.R., and Hardouin, L., On the control of max-plus linear system subject to state restriction. *Automatica*, 47 (5), 988–992 (2011).
- [14] Amari, S., Demongodin, I., Loiseau J.J., and Martinez, C., Max-plus control design for temporal constraints meeting in timed event graphs. *IEEE Transactions on Automatic Control*, 57 (2), 462–467(2012)
- [15] Houssin, L., Lahaye, S., and Boimond, J.L., Control of (max,+)-linear systems minimizing delays, *Discrete Event Dynamic Systems*, 2012, DOI: 10.1007/s10626-012-0143-1
- [16] Declerck, P. and Guezzi, A., Predictive control of timed event graphs with specifications defined by P-time event graphs, *Discrete Event Dynamic Systems*, 2012, DOI: 10.1007/s10626-012-0150-2
- [17] Van den Boom, T.J.J. and De Schutter, B., Model predictive control for perturbed max-plus-linear systems: a stochastic approach. *International Journal of Control*, 77 (3), 302–309 (2004).
- [18] Necoara, I., Van den Boom, T.J.J., and De Schutter, B., Hellendoorn, H., Stabilization of max-plus-linear systems using model predictive control: the unconstrained case, *Automatica*, 44 (4), 971–981, (2008).
- [19] Goto, H. and Takahashi, H., Efficient representation of the state equation in max-plus linear systems with interval constrained parameters. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E95-A (2), 608–612 (2012).
- [20] Goto, H., Dual representation and its online scheduling method for event-varying DESs with capacity constraints. *International Journal of Control*, 81 (4), 651–660 (2008).

- [21] Goto, H., A lightweight model predictive controller for repetitive discrete event systems, Asian Journal of Control, 2013 (in press)
- [22] Nocedal, J. and Wright, S., Numerical Optimization Second Edition. Springer, New York (1999).
- [23] Hillier, F. S. and Lieberman, G. J., Introduction to Operations Research. McGraw Hill, New York (2004).
- Hiroyuki Goto** is a professor at the department of industrial & system engineering, Hosei University, Japan. He received his B.S. and M.S. degrees from The University of Tokyo in 1995 and 1997, respectively. He received his D.E. degree from Tokyo Metropolitan Institute of Technology in 2004. His research interests include controller design for discrete event systems and high-performance computing.*