2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Block Diagram-Based Modeling of Manufacturing Systems Using Max-Plus Algebra

Aleksey Imaev
School of Electrical Engineering
and Computer Science
Ohio University
Athens, OH 45701, USA
ai400695@ohio.edu

Robert P. Judd
Cooper Industries Professor and Chair
Industrial & Manufacturing Systems
Engineering
Ohio University
Athens, OH 45701, USA
judd@ohio.edu

***Abstract*— A novel approach to modeling of manufacturing systems is proposed. The model is based on block diagrams. The block can be a single machine, a single part or a factory. The interconnection of blocks in the block diagram correspond to routing of parts and resources through the manufacturing subsystems. The model is hierarchial – a network of blocks can be combined into one block that has the same input/output structure. The model is described using max-plus algebra, which provides means for calculating system's performance characteristics.**

I. INTRODUCTION

A Discrete Event System (DES) is a system, which is characterized by a set of states and a set of events [1]. The set of events cause the DES to change its state at discrete time intervals. This work focuses on discrete event manufacturing systems. Typical events occurring in a manufacturing system include arrival of a part, completion of a finished part and removal of a finished part from the system. A deterministic manufacturing system is the one which is conflict- or choice free [2], therefore it is assumed that the routes of all the parts are established, the sequences of parts on the machines are known and the processing times are fixed.

DES theory is mainly concerned with creating DES models. The existing modeling frameworks for deterministic manufacturing systems include discrete event simulation, timed event graphs (TEGs), timed automata, queueing networks and max-plus algebra [3]. Discrete event simulation (e.g. [4]) does not supply equations needed to analyze system's behavior. Performance characteristics of TEG and directed graph models can be analyzed using path-based approaches (e.g. [5], [6], [7]) and integer/linear programming approaches (e.g. [8], [9], [10]). Queueing networks are usually used to evaluate long term performance characteristics of *stochastic* manufacturing systems, with the exception of the max-plus linear queueing networks [11].

Max-plus algebra is an attractive tool for modeling of manufacturing systems because the event timing dynamics of any deterministic manufacturing system can be expressed by a set of linear equations in the max-plus algebra. It provides computational engine for calculating system's quantitative characteristics. Fundamentally, the event timing equations in timed event graphs or max-plus linear queueing networks can always be written in terms of the max-plus algebra (e.g. [12], [13], [14], [15], [11]). Furthermore, the event timing equations for a manufacturing system may be obtained directly from system's specifications using the approach proposed by Doustmohammadi and Kamen [16].

The paper presents a novel modeling approach for deterministic manufacturing systems. A block diagram type of model is proposed. A manufacturing system is represented as a network of subsystems. Each subsystem is modeled as a block with three inputs and three outputs. The block can be a single machine, a single part or a factory. The model is hierarchial – a network of blocks can be combined into one block that has the same input/output structure. The blocks in the block diagram are interconnected through a) part-flow interconnections, which specify flow of parts through the diagram, and b) resource-flow interconnections, which specify flow of resources through the diagram. The model is expressed as a system of linear event-timing equations in max-plus algebra. The proposed model is a generalization and extension of the approach recently presented by Imaev and Judd [17]. The main difference between [17] and this paper is that in this paper the model has three inputs and three outputs, rather than two inputs and two outputs, which greatly improves flexibility of the model.

II. MAX-PLUS ALGEBRA BASICS

This section provides a brief review of the max-plus algebra and introduces some of the notation used throughout the paper. A comprehensive review of the max plus algebra can be found in [18], [13].

Define $\varepsilon = -\infty$ and $\mathbb{R}_{\max} = \{\mathbb{R} \cup \varepsilon\}$, where $\mathbb{R}$ is the set of real numbers. The two max-plus algebraic operations, $\oplus$ and $\otimes$, are defined as follows:

$$a \oplus b = \max(a, b) \quad a \otimes b = a + b,$$

for elements $a, b \in \mathbb{R}_{\max}$. Operation $\oplus$ has null element, ε, since $a \oplus \varepsilon = a$. Similarly operation $\otimes$ has unit element, $e = 0$, as $a \otimes e = a$.

Max plus algebra is extended to matrices in the same way as conventional algebra but with + replaced by $\oplus$ and $\times$ replaced by $\otimes$. A set of all $n \times m$ matrices is denoted by

$\mathbb{R}_{\max}^{m\times n}$. We say that an $n \times m$ matrix $\mathbf{A}$ exists if and only if $\mathbf{A} \in \mathbb{R}_{\max}^{n\times m}$.

Analogous to conventional algebra $\otimes$ is assumed precedence over $\oplus$ and if it is clear that the $\otimes$ symbol is used it is sometimes omitted, i.e. $\mathbf{A} \oplus \mathbf{BC}$ should be understood as $\mathbf{A} \oplus (\mathbf{B} \otimes \mathbf{C})$.

For any square matrix $\mathbf{A} \in \mathbb{R}_{\max}^{n\times n}$, $\mathbf{A}$ in n^{th} power is defined by

$$\mathbf{A}^{\otimes n} = \mathbf{A} \otimes \mathbf{A} \otimes \ldots \mathbf{A} \rightarrow n \text{ times}.$$

Define Kleen star operator on $\mathbf{A}$ denoted by $\mathbf{A}^*$ as

$$\mathbf{A}^* = \mathbf{A}^{\otimes e} \oplus \mathbf{A}^{\otimes 1} \oplus \ldots \oplus \mathbf{A}^{\otimes \infty} = \bigoplus_{k=0}^{k=\infty} \mathbf{A}^{\otimes k},$$

where $\mathbf{A}^{\otimes e} = \mathbf{E}$ and $\mathbf{E} \in \mathbb{R}_{\max}^{n\times n}$ refers to identity matrix which has e's on the main diagonal and ε's elsewhere. $\mathbf{A}^*$ can be computed in at most $O(n^3)$ time using the Floyd-Warshall algorithm [19].

Theorem 2.1: [18, Theorem 2.10] $\mathbf{x} = \mathbf{A}^* \otimes \mathbf{b}$ solves the equation $\mathbf{x} = \mathbf{A} \otimes \mathbf{x} \oplus \mathbf{b}$, provided that $\mathbf{A}^*$ exists.

For a positive integer K, define $\underline{K} = \{1, 2, \ldots, K\}$. Let $\mathbf{s}$ be an ordered set or a vector. Then $|\mathbf{s}|$ gives the number of elements in $\mathbf{s}$. The i-th element of $\mathbf{s}$ is denoted by $[\mathbf{s}]_i$, for any $i \in \underline{|\mathbf{s}|}$.

III. General Modeling Block of a Manufacturing System

Consider a manufacturing system. In order to operate the system requires a set of parts and a set of resources. After the system is done with the parts and the resources, they are are released by the system. Let $\mathbf{m}$ denote an ordered set of system's resources, such as machines, buffers, etc. Let $\mathbf{n}^{in}$ be the ordered set of parts that enter the system and let $\mathbf{n}^{out}$ be the ordered set of parts that leave the system. For $k \in \{1, 2, \ldots, |\mathbf{m}|\}$, let $[\mathbf{m}]_k$ denote the k-th resource in the set $\mathbf{m}$. Similarly, for $i \in \{1, 2, \ldots, |\mathbf{n}^{in}|\}$ and $j \in \{1, 2, \ldots, |\mathbf{n}^{out}|\}$, let $[\mathbf{n}^{in}]_i$ and $[\mathbf{n}^{out}]_j$ denote the i-th part in $\mathbf{n}^{in}$ and j-th part in $\mathbf{n}^{out}$, respectively. If the manufacturing process involves part assembly or disassembly then $\mathbf{n}^{in} \neq \mathbf{n}^{out}$ because during assembly several parts are needed to create a new part and during disassembly a single part is disassembled into several new parts. If there are no assembly and disassembly machines in the system then we can set $\mathbf{n}^{in} = \mathbf{n}^{out} = \mathbf{n}$ as the order of elements in either $\mathbf{m}$, $\mathbf{n}^{in}$ or $\mathbf{n}^{out}$ can be chosen arbitrary.

The system can be modeled by a block with three inputs and three outputs. The inputs, $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ are defined as

- $[\mathbf{u}]_i$ is the time when $[\mathbf{n}^{in}]_i$ becomes available for the system;
- $[\mathbf{v}]_j$ is the time when $[\mathbf{n}^{out}]_j$ is removed from the system;
- $[\mathbf{w}]_k$ is the time when $[\mathbf{m}]_k$ becomes available for the system.

The outputs, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ are defined as:

- $[\mathbf{x}]_j$ is the time when $[\mathbf{n}^{out}]_j$ is ready to leave the system;

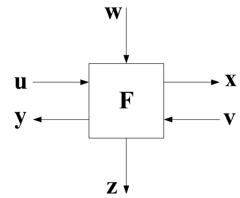


Fig. 1. Block representation of a system

- $[\mathbf{y}]_i$ is the time when $[\mathbf{n}^{in}]_i$ actually enters the system;
- $[\mathbf{z}]_k$ is the time when $[\mathbf{m}]_k$ is "set free" by the system.

It can be seen that input and output variables are defined with respect to $\mathbf{m}$, $\mathbf{n}^{in}$ and $\mathbf{n}^{out}$. In particular, $\mathbf{m}$ is associated $\mathbf{w}$, $\mathbf{z}$; $\mathbf{n}^{in}$ is associated with $\mathbf{u}$, $\mathbf{y}$; and $\mathbf{n}^{out}$ is associated with $\mathbf{x}$ and $\mathbf{v}$.

It is assumed that the system is deterministic, i.e. the routing of parts through the resources, the processing order of parts on the resources and the processing times of parts on the resources are known and fixed. Then its output can be described in terms of its input by the following equation in the max-plus algebra

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\\ \mathbf{z}\end{bmatrix} = \begin{bmatrix}\mathbf{F}_{xu} & \mathbf{F}_{xv} & \mathbf{F}_{xw}\\ \mathbf{F}_{yu} & \mathbf{F}_{yv} & \mathbf{F}_{yw}\\ \mathbf{F}_{zu} & \mathbf{F}_{zv} & \mathbf{F}_{zw}\end{bmatrix} \begin{bmatrix}\mathbf{u}\\ \mathbf{v}\\ \mathbf{w}\end{bmatrix} = \mathbf{F} \begin{bmatrix}\mathbf{u}\\ \mathbf{v}\\ \mathbf{w}\end{bmatrix}, \quad (1)$$

where $\mathbf{F}$ is a matrix that describes input-output relation - it is called the system matrix. Equation (1) along with definitions of $\mathbf{m}$, $\mathbf{n}^{in}$ and $\mathbf{n}^{out}$ provide a modeling abstraction of any deterministic manufacturing system by means of block diagram having three inputs and three outputs as shown in Figure 1.

IV. Composition of Blocks

A manufacturing system can be represented as a network of subsystems where each subsystem is modeled as described in the previous section. Let S_c be a system composed from a set of M manufacturing subsystems $\{S_1, S_2, \ldots, S_M\}$. Let $\mathbf{m}_c$, $\mathbf{n}_c^{in}$, $\mathbf{n}_c^{out}$ be ordered sets of resources and parts associated with S_c. Let the inputs and the outputs of S_c, namely $\mathbf{u}_c$, $\mathbf{v}_c$, $\mathbf{w}_c$ and $\mathbf{x}_c$, $\mathbf{y}_c$, $\mathbf{z}_c$, be defined with respect to $\mathbf{m}_c$, $\mathbf{n}_c^{in}$, $\mathbf{n}_c^{out}$.

Each subsystem $S_{i\in\underline{M}}$ is represented by an equation of the form (1) or, specifically,

$$\begin{bmatrix}\mathbf{x}_i\\ \mathbf{y}_i\\ \mathbf{z}_i\end{bmatrix} = \begin{bmatrix}\mathbf{F}_{xu,i} & \mathbf{F}_{xv,i} & \mathbf{F}_{xw,i}\\ \mathbf{F}_{yu,i} & \mathbf{F}_{yv,i} & \mathbf{F}_{yw,i}\\ \mathbf{F}_{zu,i} & \mathbf{F}_{zv,i} & \mathbf{F}_{zw,i}\end{bmatrix} \begin{bmatrix}\mathbf{u}_i\\ \mathbf{v}_i\\ \mathbf{w}_i\end{bmatrix} = \mathbf{F}_i \begin{bmatrix}\mathbf{u}_i\\ \mathbf{v}_i\\ \mathbf{w}_i\end{bmatrix}, \quad (2)$$

where inputs and outputs of S_i are defined with respect to ordered set of resources, $\mathbf{m}_i$, and ordered sets of parts, namely $\mathbf{n}_i^{in}$ and $\mathbf{n}_i^{out}$.

The subsystems $S_1, S_2, \ldots, S_M$ – all share the system's parts and resources. It is assumed that there are no delays associated with transportation of parts or resources from S_j to S_i - rather these delays can always be modeled by an appropriate manufacturing block or as part of S_i or S_j.

Consider a resource m, which is first used by S_j and then it is used by S_i. Suppose that $m = [\mathbf{m}_j]_l = [\mathbf{m}_i]_k$, where l

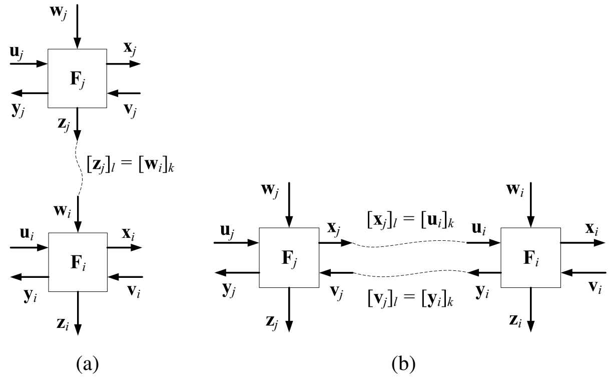


Fig. 2. Interconnection of blocks: (a) describes resource-flow interconnection when $[\mathbf{m}_j]_l \to [\mathbf{m}_i]_k$; and, (b) describes part-flow interconnection when $[\mathbf{n}_j^{out}]_l \to [\mathbf{n}_i^{in}]_k$.

and k point to the location of m in $\mathbf{m}_j$ and $\mathbf{m}_i$, respectively. It is said that $[\mathbf{m}_j]_l$ is routed to $[\mathbf{m}_i]_k$, which is denoted by $[\mathbf{m}_j]_l \to [\mathbf{m}_i]_k$. Resource m is available to S_i after S_j is done using m, therefore $[\mathbf{w}_i]_k = [\mathbf{z}_j]_l$, as shown in Figure 2(a).

Likewise, consider part n which enters system S_i from an upstream system S_j. Suppose that $n = [\mathbf{n}_j^{out}]_l = [\mathbf{n}_i^{in}]_k$, where indexes l and k point to the location of n in $\mathbf{n}_j^{out}$ and $\mathbf{n}_i^{in}$, respectively. It is said that $[\mathbf{n}_j^{out}]_l$ is routed to $[\mathbf{n}_i^{in}]_k$, which is denoted by $[\mathbf{n}_j^{out}]_l \to [\mathbf{n}_i^{in}]_k$. Since n becomes available to S_i at the time instance when it is ready to leave S_j, we have $[\mathbf{u}_i]_k = [\mathbf{x}_j]_l$. In addition, the part n is removed from S_j when n enters S_i, therefore $[\mathbf{v}_j]_l = [\mathbf{y}_i]_k$, as shown in Figure 2(b).

It follows that flow of *parts* through the manufacturing subsystems is represented by horizontal interconnections (e.g., Figure 2(b)). We will refer to this type of interconnections as *part-flow* interconnections. Likewise, flow of *resources* through the subsystems is represented by vertical interconnections (e.g., Figure 2(a)). We will refer to this type of interconnections as *resource-flow* interconnections.

Routing of parts and resources through the diagram is mathematically represented by means of part-flow and resource-flow interconnection matrices. Define

$$\tilde{\mathbf{n}}^{in} = \begin{bmatrix} \mathbf{n}_1^{in} \\ \mathbf{n}_2^{in} \\ \vdots \\ \mathbf{n}_M^{in} \end{bmatrix}, \quad \tilde{\mathbf{n}}^{out} = \begin{bmatrix} \mathbf{n}_1^{out} \\ \mathbf{n}_2^{out} \\ \vdots \\ \mathbf{n}_M^{out} \end{bmatrix}, \quad \tilde{\mathbf{m}} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_M \end{bmatrix}.$$

Resource-flow interconnection matrices are defined as:

$$[\mathbf{Q}_{in}]_{i \in \underline{|\tilde{\mathbf{m}}|}, j \in \underline{|\mathbf{m}_c|}} = \begin{cases} e & \text{if } [\mathbf{m}_c]_j \to [\tilde{\mathbf{m}}]_i, \\ \varepsilon & \text{otherwise;} \end{cases}$$

$$[\mathbf{Q}]_{i \in \underline{|\tilde{\mathbf{m}}|}, j \in \underline{|\tilde{\mathbf{m}}|}} = \begin{cases} e & \text{if } [\tilde{\mathbf{m}}]_j \to [\tilde{\mathbf{m}}]_i, \\ \varepsilon & \text{otherwise;} \end{cases}$$

$$[\mathbf{Q}_{out}]_{i \in \underline{|\mathbf{m}_c|}, j \in \underline{|\tilde{\mathbf{m}}|}} = \begin{cases} e & \text{if } [\tilde{\mathbf{m}}]_j \to [\mathbf{m}_c]_i, \\ \varepsilon & \text{otherwise.} \end{cases}$$

Part-flow interconnection matrices are defined as:

$$[\mathbf{R}_{in}]_{i \in \underline{|\tilde{\mathbf{n}}^{in}|}, j \in \underline{|\mathbf{n}_c^{in}|}} = \begin{cases} e & \text{if } [\mathbf{n}_c^{in}]_j \to [\tilde{\mathbf{n}}^{in}]_i, \\ \varepsilon & \text{otherwise;} \end{cases}$$

$$[\mathbf{R}]_{i \in \underline{|\tilde{\mathbf{n}}^{in}|}, j \in \underline{|\tilde{\mathbf{n}}^{out}|}} = \begin{cases} e & \text{if } [\tilde{\mathbf{n}}^{out}]_j \to [\tilde{\mathbf{n}}^{in}]_i, \\ \varepsilon & \text{otherwise;} \end{cases}$$

$$[\mathbf{R}_{out}]_{i \in \underline{|\mathbf{n}_c^{out}|}, j \in \underline{|\tilde{\mathbf{n}}^{out}|}} = \begin{cases} e & \text{if } [\tilde{\mathbf{n}}^{out}]_j \to [\mathbf{n}_c^{out}]_i, \\ \varepsilon & \text{otherwise.} \end{cases}$$

The following develops formulas for composing manufacturing systems. Define vectors

$$\tilde{\mathbf{u}} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_M \end{bmatrix}, \quad \tilde{\mathbf{v}} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_M \end{bmatrix}, \quad \tilde{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_M \end{bmatrix}.$$

Similarly define $\tilde{\mathbf{x}}$, $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{z}}$. Define

$$\tilde{\mathbf{F}}_{xu} = \begin{bmatrix} \mathbf{F}_{xu,1} & \varepsilon & & \varepsilon \\ \varepsilon & \mathbf{F}_{xu,2} & & \varepsilon \\ & & \ddots & \\ \varepsilon & \varepsilon & & \mathbf{F}_{xu,M} \end{bmatrix}.$$

Similarly define $\tilde{\mathbf{F}}_{xv}$, $\tilde{\mathbf{F}}_{xw}$, $\tilde{\mathbf{F}}_{yu}$, etc. Then

$$\begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{y}} \\ \tilde{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{F}}_{xu} & \tilde{\mathbf{F}}_{xv} & \tilde{\mathbf{F}}_{xw} \\ \tilde{\mathbf{F}}_{yu} & \tilde{\mathbf{F}}_{yv} & \tilde{\mathbf{F}}_{yw} \\ \tilde{\mathbf{F}}_{zu} & \tilde{\mathbf{F}}_{zv} & \tilde{\mathbf{F}}_{zw} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{v}} \\ \tilde{\mathbf{w}} \end{bmatrix} = \tilde{\mathbf{F}} \begin{bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{v}} \\ \tilde{\mathbf{w}} \end{bmatrix}. \tag{3}$$

From the definition of the interconnection matrices it follows

$$\begin{aligned} \tilde{\mathbf{u}} &= \mathbf{R}\tilde{\mathbf{x}} \oplus \mathbf{R}_{in}\mathbf{u}_c, \\ \tilde{\mathbf{v}} &= \mathbf{R}^T\tilde{\mathbf{y}} \oplus \mathbf{R}_{out}^T\mathbf{v}_c, \\ \tilde{\mathbf{w}} &= \mathbf{Q}\tilde{\mathbf{z}} \oplus \mathbf{Q}_{in}\mathbf{w}_c. \end{aligned} \tag{4}$$

Outputs of S_c can be expressed as

$$\begin{aligned} \mathbf{x}_c &= \mathbf{R}_{out}\tilde{\mathbf{x}}, \\ \mathbf{y}_c &= \mathbf{R}_{in}^T\tilde{\mathbf{y}}, \\ \mathbf{z}_c &= \mathbf{Q}_{out}\tilde{\mathbf{z}}. \end{aligned} \tag{5}$$

Equations (4) can be written as

$$\begin{aligned} \begin{bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{v}} \\ \tilde{\mathbf{w}} \end{bmatrix} &= \begin{bmatrix} \mathbf{R} & \varepsilon & \varepsilon \\ \varepsilon & \mathbf{R}^T & \varepsilon \\ \varepsilon & \varepsilon & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{y}} \\ \tilde{\mathbf{z}} \end{bmatrix} \\ &\oplus \begin{bmatrix} \mathbf{R}_{in} & \varepsilon & \varepsilon \\ \varepsilon & \mathbf{R}_{out}^T & \varepsilon \\ \varepsilon & \varepsilon & \mathbf{Q}_{in} \end{bmatrix} \begin{bmatrix} \mathbf{u}_c \\ \mathbf{v}_c \\ \mathbf{w}_c \end{bmatrix}. \end{aligned} \tag{6}$$

Equations (5) can be written as

$$\begin{bmatrix} \mathbf{x}_c \\ \mathbf{y}_c \\ \mathbf{z}_c \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{out} & \varepsilon & \varepsilon \\ \varepsilon & \mathbf{R}_{in}^T & \varepsilon \\ \varepsilon & \varepsilon & \mathbf{Q}_{out} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{y}} \\ \tilde{\mathbf{z}} \end{bmatrix}. \tag{7}$$

Substituting (6) into (3) we obtain

$$\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{y}}\\ \tilde{\mathbf{z}}\end{bmatrix} = \tilde{\mathbf{F}}\begin{bmatrix}\mathbf{R} & \varepsilon & \varepsilon\\ \varepsilon & \mathbf{R}^T & \varepsilon\\ \varepsilon & \varepsilon & \mathbf{Q}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{y}}\\ \tilde{\mathbf{z}}\end{bmatrix} \oplus \tilde{\mathbf{F}}\begin{bmatrix}\mathbf{R}_{in} & \varepsilon & \varepsilon\\ \varepsilon & \mathbf{R}_{out}^T & \varepsilon\\ \varepsilon & \varepsilon & \mathbf{Q}_{in}\end{bmatrix}\begin{bmatrix}\mathbf{u}_c\\ \mathbf{v}_c\\ \mathbf{w}_c\end{bmatrix}. \quad (8)$$

From Theorem 2.1 it follows that

$$\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{y}}\\ \tilde{\mathbf{z}}\end{bmatrix} = \left(\tilde{\mathbf{F}}\begin{bmatrix}\mathbf{R} & \varepsilon & \varepsilon\\ \varepsilon & \mathbf{R}^T & \varepsilon\\ \varepsilon & \varepsilon & \mathbf{Q}\end{bmatrix}\right)^* \otimes \tilde{\mathbf{F}}\begin{bmatrix}\mathbf{R}_{in} & \varepsilon & \varepsilon\\ \varepsilon & \mathbf{R}_{out}^T & \varepsilon\\ \varepsilon & \varepsilon & \mathbf{Q}_{in}\end{bmatrix}\begin{bmatrix}\mathbf{u}_c\\ \mathbf{v}_c\\ \mathbf{w}_c\end{bmatrix} \quad (9)$$

Substituting (9) into (7) we get

$$\begin{bmatrix}\mathbf{x}_c\\ \mathbf{y}_c\\ \mathbf{z}_c\end{bmatrix} = \begin{bmatrix}\mathbf{R}_{out} & \varepsilon & \varepsilon\\ \varepsilon & \mathbf{R}_{in}^T & \varepsilon\\ \varepsilon & \varepsilon & \mathbf{Q}_{out}\end{bmatrix}\left(\tilde{\mathbf{F}}\begin{bmatrix}\mathbf{R} & \varepsilon & \varepsilon\\ \varepsilon & \mathbf{R}^T & \varepsilon\\ \varepsilon & \varepsilon & \mathbf{Q}\end{bmatrix}\right)^* \otimes \tilde{\mathbf{F}}\begin{bmatrix}\mathbf{R}_{in} & \varepsilon & \varepsilon\\ \varepsilon & \mathbf{R}_{out}^T & \varepsilon\\ \varepsilon & \varepsilon & \mathbf{Q}_{in}\end{bmatrix}\begin{bmatrix}\mathbf{u}_c\\ \mathbf{v}_c\\ \mathbf{w}_c\end{bmatrix} \quad (10)$$

Equation (10) gives general expression for the system matrix of S_c. This proves that any composition of systems represented by (1) results in a system that is also represented by (1).

V. Deadlock Detection

Consider system S_c, modeled by matrix $\mathbf{F}_c$, which is defined with respect to $\mathbf{n}_c^{in}$, $\mathbf{n}_c^{out}$ and $\mathbf{m}_c$. If the system has deadlocks than some jobs that enter the system will never be able to leave it. Suppose that job $n = [\mathbf{n}_c^{out}]_i$ is in deadlock and cannot leave the system, then $[\mathbf{x}_c]_i = +\infty$. This means that $\mathbf{F}_c$ contains elements that are equal to $+\infty$; in other words $\mathbf{F}_c$ will not exist. On the contrary, if $\mathbf{F}_c$ exists then the system is free of deadlocks.

Suppose that S_c is a network of subsystems S_i, for $i = 1, 2, \ldots, M$, such that each S_i is deadlock free. Then from (10) it follows that $\mathbf{F}$ exists (and, therefore, S_c is deadlock free) if and only if the following exists:

$$\left(\tilde{\mathbf{F}}_c\begin{bmatrix}\mathbf{R} & \varepsilon & \varepsilon\\ \varepsilon & \mathbf{R}^T & \varepsilon\\ \varepsilon & \varepsilon & \mathbf{Q}\end{bmatrix}\right)^*. \quad (11)$$

VI. Basic Blocks

In this sections timing models of basic manufacturing blocks are presented. All models share the generic structure described by (1) with three inputs and three outputs.

A. Single machine processing single part

Consider machine m processing part n. Let t be processing time of n on m. Suppose that the system is modeled by using equation of the form (1) having inputs u, v, w and outputs x, y, z, which are all scalars because there is only one resource and one part.

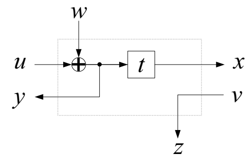
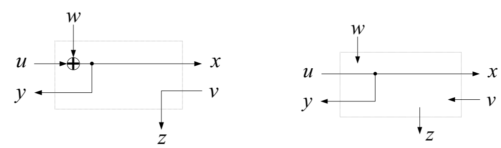


Fig. 3. Single machine manufacturing single part.



(a) Unit capacity buffer (b) Unlimited capacity buffer with random access

Fig. 4. Buffer models

The part n enters the system as soon as both m and n are available, therefore

$$y = u \oplus w.$$

The part is ready to leave the system as soon as its processing is done on the machine, therefore

$$x = t(u \oplus w).$$

The machine is ”set free” by the system as soon as n is removed from the system, therefore

$$z = v.$$

Thus, we have

$$\begin{bmatrix}x\\ y\\ z\end{bmatrix} = \begin{bmatrix}t & \varepsilon & t\\ e & \varepsilon & e\\ \varepsilon & e & \varepsilon\end{bmatrix}\begin{bmatrix}u\\ v\\ w\end{bmatrix}. \quad (12)$$

Block diagram model of the system is provided in Figure 3.

B. Unit capacity buffer

McCormick et al. [6] show that a buffer of unit capacity can be represented by a resource having zero processing time for jobs that enter the buffer. Therefore for buffer of unit capacity (12) becomes

$$\begin{bmatrix}x\\ y\\ z\end{bmatrix} = \begin{bmatrix}e & \varepsilon & e\\ e & \varepsilon & e\\ \varepsilon & e & \varepsilon\end{bmatrix}\begin{bmatrix}u\\ v\\ w\end{bmatrix}, \quad (13)$$

because $t = e$. Block diagram representation of (14) is provided in Figure 4(a).

C. Random access buffer with infinite capacity

Consider random access buffer with unlimited capacity for storing parts. The buffer is always available to accept parts because of its unlimited capacity, therefore $w = \varepsilon$ and $z = \varepsilon$. The part enters the buffer as soon as it becomes available to the buffer, therefore $y = u$. Also, the part is ready to leave the buffer as soon as it entered the buffer, therefore $x = y = u$. Hence,

$$\begin{bmatrix}x\\ y\\ z\end{bmatrix} = \begin{bmatrix}e & \varepsilon & \varepsilon\\ e & \varepsilon & \varepsilon\\ \varepsilon & \varepsilon & \varepsilon\end{bmatrix}\begin{bmatrix}u\\ v\\ w\end{bmatrix}. \quad (14)$$

Block diagram representation of the model is shown in Figure 4(b)

VII. Line Applications

Line applications include models of a single part processed by a set of M resources and models of a single resource processing a set of N parts. They are called line applications because in the former case basic blocks are stacked horizontally and in the latter case basic blocks are stacked vertically.

A. Unit capacity machine processing a set of parts

Consider a unit capacity machine m processing a set of parts $\mathbf{n} = [n_1, n_2, \dots n_N]$ in the order specified by $\mathbf{n}$. Suppose that m is neither assembly or disassembly machine, hence $\mathbf{n} = \mathbf{n}_{in} = \mathbf{n}_{out}$. The model of the system is denoted by S_c with inputs and outputs $\mathbf{u}_c$, $\mathbf{v}_c$, w_c and $\mathbf{x}_c$, $\mathbf{y}_c$, z_c that are defined w.r.t. $\mathbf{m}_c = m$, $\mathbf{n}_c^{in} = \mathbf{n}_c^{out} = \mathbf{n}$.

The system S_c can be modeled by a sequence of N blocks $S_{i\in\underline{N}}$ stacked vertically as shown in Figure 5(a), where each S_i is a basic block represented by an equation of the form (12), i.e.

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} t_i & \varepsilon & t_i \\ e & \varepsilon & e \\ \varepsilon & e & \varepsilon \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = \mathbf{F}_i \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix}, \tag{15}$$

where t_i is the processing time of n_i on m.

The blocks in the block diagram are interconnected only through the resource-flow type of interconnections. We have $w_i = z_{i-1}$ for $i > 1$. Therefore the interconnection matrices for the system take the following form

$$\mathbf{Q}_{in} = \mathbf{J}, \quad \mathbf{Q} = \mathbf{H}, \quad \mathbf{Q}_{out} = \mathbf{G}, \tag{16}$$

$$\mathbf{R}_{in} = \mathbf{E}, \quad \mathbf{R} = \varepsilon, \quad \mathbf{R}_{out} = \mathbf{E}, \quad \text{where} \tag{17}$$

$$\mathbf{J} = \begin{bmatrix} e \\ \varepsilon \\ \vdots \\ \varepsilon \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \varepsilon & & \varepsilon & \varepsilon \\ e & & \varepsilon & \varepsilon \\ & \ddots & & \\ \varepsilon & & e & \varepsilon \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \varepsilon & \dots & \varepsilon & e \end{bmatrix}.$$

From (3) utilizing equation (15) we obtain

$$\begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{y}} \\ \tilde{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{P} & \varepsilon & \mathbf{P} \\ \mathbf{E} & \varepsilon & \mathbf{E} \\ \varepsilon & \mathbf{E} & \varepsilon \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{v}} \\ \tilde{\mathbf{w}} \end{bmatrix}, \tag{18}$$

where

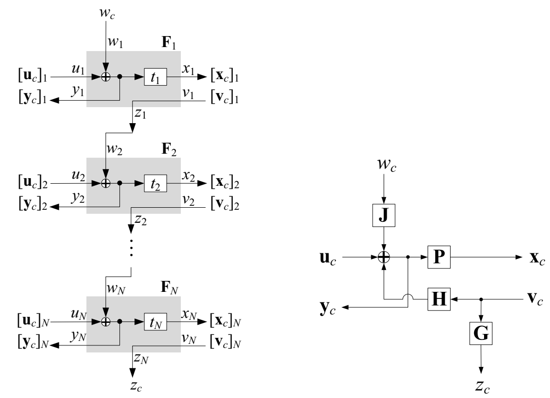
$$\mathbf{P} = \begin{bmatrix} t_1 & \varepsilon & & \varepsilon \\ \varepsilon & t_2 & & \varepsilon \\ & & \ddots & \\ \varepsilon & \varepsilon & & t_N \end{bmatrix}.$$

Therefore

$$\begin{aligned} \tilde{\mathbf{x}} &= \mathbf{P}\tilde{\mathbf{u}} \oplus \mathbf{P}\tilde{\mathbf{w}} \\ \tilde{\mathbf{y}} &= \tilde{\mathbf{u}} \oplus \tilde{\mathbf{w}} \\ \tilde{\mathbf{z}} &= \tilde{\mathbf{v}}. \end{aligned} \tag{19}$$

Substituting (16) and (17) into (4) and (5) we get

$$\begin{aligned} \tilde{\mathbf{u}} &= \mathbf{u}_c, & \mathbf{x}_c &= \tilde{\mathbf{x}}, \\ \tilde{\mathbf{v}} &= \mathbf{v}_c, & \mathbf{y}_c &= \tilde{\mathbf{y}}, \\ \tilde{\mathbf{w}} &= \mathbf{H}\tilde{\mathbf{z}} \oplus \mathbf{J}w_c & z_c &= \mathbf{G}\tilde{\mathbf{z}}. \end{aligned} \tag{20}$$



(a) Vertical stack of blocks – each $\mathbf{F}_i$ is modeled by (15).

(b) Compact representation of the system. Refer to (21).

Fig. 5. Model of single machine processing set of parts

From (19) and (20) it follows

$$\begin{aligned} \tilde{\mathbf{z}} &= \tilde{\mathbf{v}} = \mathbf{v}_c, \\ \mathbf{y}_c &= \tilde{\mathbf{u}} \oplus \tilde{\mathbf{w}} = \mathbf{u}_c \oplus \mathbf{H}\tilde{\mathbf{z}} \oplus \mathbf{J}w_c = \mathbf{u}_c \oplus \mathbf{H}\mathbf{v}_c \oplus \mathbf{J}w_c, \\ \mathbf{x}_c &= \mathbf{P}\tilde{\mathbf{y}} = \mathbf{P}\mathbf{u}_c \oplus \mathbf{P}\mathbf{H}\mathbf{v}_c \oplus \mathbf{P}\mathbf{J}w_c, \\ z_c &= \mathbf{G}\tilde{\mathbf{z}} = \mathbf{G}\mathbf{v}_c. \end{aligned}$$

Therefore

$$\begin{bmatrix} \mathbf{x}_c \\ \mathbf{y}_c \\ z_c \end{bmatrix} = \begin{bmatrix} \mathbf{P} & \mathbf{PH} & \mathbf{PJ} \\ \mathbf{E} & \mathbf{H} & \mathbf{J} \\ \varepsilon & \mathbf{G} & \varepsilon \end{bmatrix} \begin{bmatrix} \mathbf{u}_c \\ \mathbf{v}_c \\ w_c \end{bmatrix}. \tag{21}$$

Equation (21) models unit capacity machine processing a set of parts. Its block diagram representation is provided in Figure 5(b).

B. Single part processed by a set of machines

Consider part n which is processed by a set of unit capacity machines $\mathbf{m} = [m_1, m_2, \dots m_M]$ in the order specified by $\mathbf{m}$. The model of the system is denoted by S_c with inputs and outputs u_c, v_c, $\mathbf{w}_c$, x_c, y_c, $\mathbf{z}_c$ that are defined w.r.t. $\mathbf{m}_c = \mathbf{m}$, $\mathbf{n}_c^{in} = \mathbf{n}_c^{out} = n$.

The system S_c can be modeled by a sequence of M blocks $S_{i\in\underline{M}}$ stacked horizontally as shown in Figure 5(a), where each S_i is a basic block represented by an equation of the form (12), i.e.

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} t_i & \varepsilon & t_i \\ e & \varepsilon & e \\ \varepsilon & e & \varepsilon \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = \mathbf{F}_i \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix}, \tag{22}$$

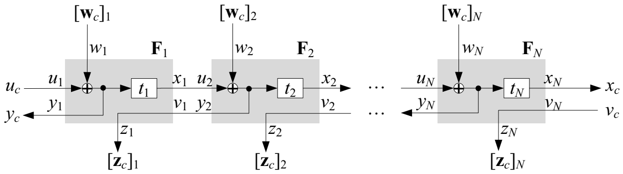
where t_i is the processing time of n on m_i.

The blocks in the block diagram are interconnected only through the part-flow type of interconnections. We have $\mathbf{u}_i = \mathbf{x}_{i-1}$ and $\mathbf{v}_{i-1} = \mathbf{y}_i$, for $i > 1$, as shown in Figure 6(a). Therefore the interconnection matrices for the system are

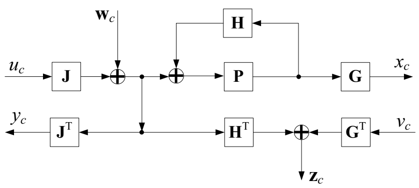
$$\mathbf{Q}_{in} = \mathbf{E}, \quad \mathbf{Q} = \varepsilon, \quad \mathbf{Q}_{out} = \mathbf{E}, \tag{23}$$

$$\mathbf{R}_{in} = \mathbf{J}, \quad \mathbf{R} = \mathbf{H}, \quad \mathbf{R}_{out} = \mathbf{G}, \quad \text{where} \tag{24}$$

where $\mathbf{J}$, $\mathbf{H}$ and $\mathbf{G}$ are defined as in the previous subsection.



(a) Horizontal stack of blocks – each $\mathbf{F}_i$ is modeled by (22).



(b) Compact representation of the system.

Fig. 6. Single part processed by set of machines

From (22) and (3) we have

$$\begin{aligned} \tilde{\mathbf{x}} &= \mathbf{P}\tilde{\mathbf{u}} \oplus \mathbf{P}\tilde{\mathbf{w}} \\ \tilde{\mathbf{y}} &= \tilde{\mathbf{u}} \oplus \tilde{\mathbf{w}} \\ \tilde{\mathbf{z}} &= \tilde{\mathbf{v}}, \end{aligned} \tag{25}$$

where

$$\mathbf{P} = \begin{bmatrix} t_1 & \varepsilon & & \varepsilon \\ \varepsilon & t_2 & & \varepsilon \\ & & \ddots & \\ \varepsilon & \epsilon & & t_M \end{bmatrix}.$$

Substituting (23) and (24) into (4) and (5) we obtain

$$\begin{aligned} &\tilde{\mathbf{u}} = \mathbf{H}\tilde{\mathbf{x}} \oplus \mathbf{J}\mathbf{u}_c, && \mathbf{x}_c = \mathbf{G}\tilde{\mathbf{x}}, \\ &\tilde{\mathbf{v}} = \mathbf{H}^T\tilde{\mathbf{y}} \oplus \mathbf{G}^T\mathbf{v}_c, && \mathbf{y}_c = \mathbf{J}^T\tilde{\mathbf{y}}, \\ &\tilde{\mathbf{w}} = \mathbf{w}_c, && \mathbf{z}_c = \tilde{\mathbf{z}}. \end{aligned} \tag{26}$$

It follows from (25) and (26) that

$$\begin{bmatrix} \mathbf{x}_c \\ \mathbf{y}_c \\ \mathbf{z}_c \end{bmatrix} = \begin{bmatrix} \mathbf{GP}(\mathbf{HP})^*\mathbf{J} & \varepsilon & \mathbf{GP}(\mathbf{HP})^* \\ \mathbf{J}^T(\mathbf{HP})^*\mathbf{J} & \varepsilon & \mathbf{J}^T(\mathbf{HP})^* \\ \mathbf{H}^T(\mathbf{HP})^*\mathbf{J} & \mathbf{G}^T & \mathbf{H}^T(\mathbf{HP})^* \end{bmatrix} \begin{bmatrix} \mathbf{u}_c \\ \mathbf{v}_c \\ \mathbf{w}_c \end{bmatrix}. \tag{27}$$

Note that $(\mathbf{HP})^*$ exists and has the following expression:

$$(\mathbf{HP})^* = \begin{bmatrix} e & \varepsilon & \dots & \varepsilon & \varepsilon \\ t_1 & e & \dots & \varepsilon & \varepsilon \\ t_1t_2 & t_2 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & e & \varepsilon \\ t_1t_2\dots t_{M-1} & t_2t_3\dots t_{M-1} & \dots & t_{M-1} & e \end{bmatrix}.$$

Block diagram of the model is shown in Figure 6(b).

VIII. Conclusion

The paper described the new approach to modeling deterministic manufacturing systems. The approach is based on block diagrams. The block can be as basic as a single machine processing a single part or as complex as a factory. The model is hierarchial - it is shown how a network of blocks can be combined into one block that has the same input output structure. The blocks in the block diagram are interconnected through part-flow and resource-flow interconnections, which allow for tracing the flow of parts and resources through the system.

The set of basic manufacturing blocks presented in this paper can be extended to include assembly and disassembly operations. Any deterministic manufacturing system, such as job-shop or flow shop, can be modeled as a network of basic manufacturing blocks, which can then be reduced into a single block.

The approach can be readily implemented in computer software because the model basically involves addition and multiplication of matrices in the max-plus algebra. The underlying max-plus equations describing the model provide means to calculating performance characteristics of the system, such as makespan, throughput, work in process, machine utilization, etc.

References

[1] B. Hruz and M.C. Zhou. *Modeling and Control of Discrete-event Dynamic Systems with Petri Nets and Other Tool*. Springer-Verlag, London, UK, 2007.

[2] M. Zhou and K. Venkatesh. *Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach*. World Scientific, 1999.

[3] Christos G. Cassandras and Stephane Lafortune. *Introduction to Discrete Event Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[4] J. Banks, J.S. Carson, B.L. Nelson, and D.M. Nicol. *Discrete-Event System Simulation*. Prentice Hall; 3 edition, 2000.

[5] R.M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23:309–311, 1978.

[6] S. Thomas McCormick, M. Pinedo, Scott Shenker, and Barry Wolf. Sequencing in an assembly line with blocking to minimize cycle time. *Oper. Res.*, 37(6):925–935, 1989.

[7] E.V. Levner. Optimal planning of parts machining on a number of machines. *Automation and Remote Control*, 12:1972–1978, 1969.

[8] H.P. Hillion and J.-M. Proth. Performance evaluation of job-shop systems using timed event-graphs. *IEEE Transactions on Automatic Control*, 34(1):3–9, Jan 1989.

[9] A. Di Febbraro, R. Minciardi, and S. Sacone. Deterministic timed event graphs for performance optimization of cyclic manufacturing processes. *IEEE Transactions on Robotics and Automation*, 13(2):169–181, Apr 1997.

[10] S. C. Ren, D. Xu, F. Wang, and M. Tan. Timed event graph-based cyclic reconfigurable flow shop modelling and optimization. *International Journal of Production Research*, 45(1):143–156, 2007.

[11] B. Heidergott. A characterisation of (max,+)-linear queueing systems. *Queueing Systems*, 35(1–4):237–262, 2000.

[12] G. Cohen, S. Gaubert, and J. Quadrat. Max-plus algebra and system theory: Where we are and where to go now. *Elsevier Annu. Rev. Control*, 23:207–219, 1999.

[13] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity*. John Wiley and Sons, West Sussex, England, 1992.

[14] Tae-Eog Lee. Stable earliest starting schedules for cyclic job shops: A linear system approach. *International Journal of Flexible Manufacturing Systems*, 12(1):59–80, 2000.

[15] N. Krivulin. The max-plus algebra approach in modelling of queueing networks. In *Summer Computer Simulation Conference*, pages 485–490, Portland, OR, July 1996.

[16] A Doustmohammadi and E.W. Kamen. Direct generation of event-timing equations for generalized flow shop systems. In *Proc. SPIE on Modeling, Simulation, and Control Technologies for Manufacturing*, volume 2596, pages 50–62, November 1995.

[17] A. Imaev and R.P. Judd. Hierarchial modeling of manufacturing systems using max-plus algebra. In *Proceedings of the American Control Conference*, June 11 - Jun 13 2008.

[18] B. Heidergott, G.J. Olsder, and J. van der Woude. *Max Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications*. Princeton University Press, 2005.

[19] R.A. Cuninghame Green. *Minimax Algebra*. Lect. Notes Econ. Math. Syst. 166. Springer, New York, 1979.