2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Hierarchial 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*— The paper describes a novel approach to model any deterministic manufacturing system. A function block diagram type of model is proposed. A block can be a single machine or a factory. The model is expressed as a system of linear event timing equations using max-plus algebra. One of the advantages of block diagram representation is the simplicity of algebraic relations between the sub-system transfer functions [1]. It is expected that the proposed model will make it possible to develop and use techniques similar to those found in classical transfer function block diagrams.**

I. INTRODUCTION

This paper describes a generic approach to model any deterministic manufacturing system. A deterministic system is the one which is conflict- or choice free [2]. Accordingly, 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. Furthermore, it is assumed that a part starts processing on a machine as soon as it enters it.

Existing modeling frameworks of deterministic manufacturing systems include discrete event simulation, timed event graphs (TEGs) and directed graphs, queueing networks and max-plus algebra [3]. Discrete event simulation (e.g. [4]) is usually computationally expensive and it doesn't supply equations needed to analyze and predict system's behavior. TEGs are a subclass of Petri nets where each place has only one incoming and one outgoing arc. Performance of TEG and directed graph models can be analyzed using path-based approaches (e.g. [5], [6], [7], [8]) and integer/linear programming approaches (e.g. [9], [10], [11]). Queueing networks are usually used to evaluate long term performance characteristics of *stochastic* manufacturing systems, with the exception of the so-called max-plus linear queueing networks [12].

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. [13], [14], [15], [16], [12]). Furthermore, the max-plus algebraic model of a manufacturing system may be obtained directly from system's specifications using the approach proposed by Doustmohammadi and Kamen [17].

What is missing in the existing modeling approaches of manufacturing systems is the block diagram type of model. It is well known that block diagrams, such as state space block diagrams and transfer function block diagrams, are widely used in control theory to model behavior of linear systems.

A novel block diagram type of model of deterministic manufacturing systems is proposed. The model is expressed as a system of linear event-timing equations in max-plus algebra. A manufacturing system is represented as a network of processing elements. Each processing element is modeled as a block with two inputs and two outputs. A block can be a single machine or a factory. Parts are routed through blocks according to system specifications. In a block diagram the interconnection of blocks with one another and with system's inputs and outputs is specified using routing matrices. Routing matrices described here are similar in concept with interconnection matrices introduced by Doustmahammadi et. al [17].

Two scheduling policies for manufacturing systems are distinguished: a conventional batch production (*static*) scheduling and a dynamical (*cyclic*) scheduling. A static production schedule tends to produce a set of required parts in one large lot. A manufacturing system that repeatedly produces an identical set of parts is called a *dynamical* or *cyclic* manufacturing system. In the paper, we limit ourselves to static deterministic manufacturing systems. Extension of the approach to dynamical systems is currently being developed.

The contributions of the paper include: a novel block diagram approach to modeling deterministic manufacturing systems, a composition methodology that reduces a network of blocks to a single block, models for machines and buffers, which constitute the basic building blocks of any manufacturing system, and the application of the approach to simple manufacturing systems such as job shops.

II. MAX-PLUS ALGEBRA BASICS

In this section an overview of the max-plus algebra is provided. For a more comprehensive review of the max plus algebra the reader is referred to [18], [14]

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

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 $a, b \in \mathbb{R}_{\max}$.

Max-plus algebra is an example of algebraic structure called *dioid*. 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

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

Define

$$\mathbf{A}^* = \bigoplus_{k=0}^{k=\infty} \mathbf{A}^{\otimes k},$$

where $\mathbf{A}^{\otimes 0} = \mathbf{E}$ 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 used in solving linear equations as shown in the next theorem.

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.

III. GENERAL MODELING BLOCK OF MANUFACTURING SYSTEM

A manufacturing system consists of a set of machines performing operations on a set of jobs. It is assumed that the sequence of jobs on machines is known as well as the order in which jobs flow through the machines. This means that the system is deterministic and therefore timings of the system's events can be expressed by linear equations in the max-plus algebra.

Let J^{in} be an ordered set of jobs that enter the system and likewise let J^{out} be an ordered set of jobs that leave the system. J^{in} and J^{out} refer to the same set of jobs, however the order of elements in J^{in} and J^{out} is arbitrary and does not have to be the same. Let $[J^{in}]_k$ and $[J^{out}]_k$ denote the k-th job in J^{in} and J^{out}, respectively. In this paper, it is assumed that $N = |J^{in}| = |J^{out}|$. [1]

The model has two inputs, $\mathbf{u}$ and $\mathbf{v}$,

- $[\mathbf{u}]_k$ is the time when job $[J^{in}]_k$ is available to the system,
- $[\mathbf{v}]_k$ is the time when job $[J^{out}]_k$ is removed from the system,

and two outputs, $\mathbf{x}$ and $\mathbf{y}$,

- $[\mathbf{x}]_k$ is the time when job $[J^{out}]_k$ is ready to leave the system,
- $[\mathbf{y}]_k$ is the time when job $[J^{in}]_k$ is entered the system,

where $k \in 1, 2, \ldots, N$.

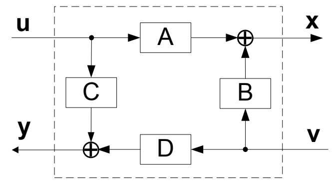

Fig. 1. Block diagram representation of a manufacturing system

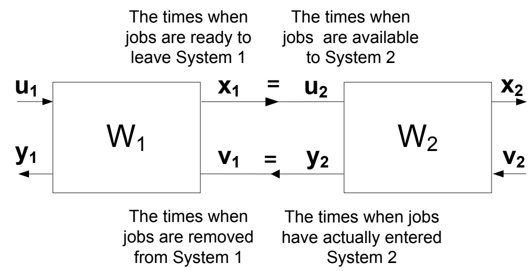

Fig. 2. Serial composition of two blocks

Since the system is max-plus linear, the output of the system can be described in terms of its input by the following equation in the max-plus algebra

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \mathbf{W} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}, \quad (1)$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are matrices in $\mathbb{R}_{\max}^{n \times n}$ that describe input-output relation and $\mathbf{W}$ is called the system matrix. The model can be graphically represented by a block diagram having two inputs and two outputs as illustrated in Figure 1.

From the definition of the model's inputs and outputs it follows that in order to have feasible model one must have

$$[\mathbf{v}]_k \geq [\mathbf{x}]_k, \quad (2)$$

as $[J^{out}]_k$ can not be removed from the system before it is actually ready to leave the system.

The definition of block's inputs and outputs facilitates a convenient method to connect blocks. Consider two subsystems modeled by $\mathbf{W}_1$ and $\mathbf{W}_2$, which are connected in serial (i.e. in flow-shop configuration). Assuming that there is no transportation delay for jobs going from System 1 to System 2, we have $\mathbf{u}_2 = \mathbf{x}_1$ and $\mathbf{v}_1 = \mathbf{y}_2$. Hence, the serial composition of the sub-systems can be represented by a block diagram shown in Figure 2.

IV. COMPOSITION OF MANUFACTURING SYSTEMS

A manufacturing system can be represented as a network of smaller subsystems where flow of jobs from one subsys-

[1] In a general manufacturing system $|J^{in}|$ and $|J^{out}|$ can be different because there may be part assembly and (or) part disassembly operations. This general case will be implemented in the future by using assembly and disassembly modeling blocks and by allowing $|J^{in}| \neq |J^{out}|$.

tem to another is specified by a routing matrix. In this section we derive equations for an aggregate system.

Let $\mathfrak{M}$ be a system composed from a set of manufacturing subsystems $\mathcal{M} = \{m_1, m_2, \ldots, m_M\}$, such that each subsystem is represented by an equation of the form (1) or, specifically,

$$\begin{bmatrix}\mathbf{x}_i\\ \mathbf{y}_i\end{bmatrix} = \begin{bmatrix}\mathbf{A}_i & \mathbf{B}_i\\ \mathbf{C}_i & \mathbf{D}_i\end{bmatrix}\begin{bmatrix}\mathbf{u}_i\\ \mathbf{v}_i\end{bmatrix}, \qquad (3)$$

for $m_i \in \mathcal{M}$. Let the inputs and the outputs of $\mathfrak{M}$ be denoted by $\mathbf{u}_{ext}$, $\mathbf{v}_{ext}$ and $\mathbf{x}_{ext}$, $\mathbf{y}_{ext}$, respectively.

Let J^{in} and J^{out} be ordered sets of jobs defined with respect to $\mathfrak{M}$. Likewise let J_i^{in} and J_i^{out} be ordered sets of jobs defined with respect to m_i. Define $N = |J^{in}| = |J^{out}|$ and $N_i = |J_i^{in}| = |J_i^{out}|$.

Next, the problem of routing jobs through systems is addressed. Consider job z which enters system m_i from an upstream system m_j. Suppose that $z \equiv [J_j^{out}]_l \equiv [J_i^{in}]_k$. Since $[J_j^{out}]_l$ and $[J_i^{in}]_k$ both refer to job z, what is the correspondence between index l and index k?

Let $\mathcal{K}_{j,i}$ be a set of jobs, which are routed from m_j to m_i. Define $\mathbf{R}_{i,j}$, where

$$[\mathbf{R}_{i,j}]_{k,l} = \begin{cases} e & \text{if } [J_j^{out}]_l \equiv [J_i^{in}]_k \in \mathcal{K}_{i,j}\\ \varepsilon & \text{otherwise,}\end{cases}$$

where $k \in \{1, 2, \ldots, N_i\}$ and $l \in \{1, 2, \ldots, N_j\}$. $\mathbf{R}_{i,j}$ is an $N_i \times N_j$ matrix describing routing of jobs from m_i to m_j.

Let $\mathcal{K}_{in,i}$ be a set of jobs routed from J^{in} to m_i. Define matrix $\mathbf{Q}_i$, where

$$[\mathbf{Q}_i]_{k,l} = \begin{cases} e & \text{if } [J^{in}]_l \equiv [J_i^{in}]_k \in \mathcal{K}_{in,i}\\ \varepsilon & \text{otherwise,}\end{cases}$$

where $k \in \{1, 2, \ldots, N\}$ and $l \in \{1, 2, \ldots, N_i\}$. $\mathbf{Q}_{i,j}$ is an $N \times N_i$ matrix describing routing of jobs from J^{in} to m_i.

Let $\mathcal{K}_{i,out}$ be a set of jobs routed from m_i to J^{out}. Define matrix $\mathbf{S}_i$, where

$$[\mathbf{S}_i]_{k,l} = \begin{cases} e & \text{if } [J_i^{out}]_l \equiv [J^{out}]_k \in \mathcal{K}_{i,out}\\ \varepsilon & \text{otherwise,}\end{cases}$$

where $k \in \{1, 2, \ldots, N_i\}$ and $l \in \{1, 2, \ldots, N\}$. $\mathbf{S}_{i,j}$ is an $N_i \times N$ matrix describing routing of jobs from m_i to J^{out}.

Define $\mathbf{R}$, such that $[\mathbf{R}]_{i,j} = \mathbf{R}_{i,j}$, where $i, j \in \{1, \ldots, M\}$. Similarly, define

$$\mathbf{Q} = \begin{bmatrix}\mathbf{Q}_1\\ \mathbf{Q}_2\\ \vdots\\ \mathbf{Q}_M\end{bmatrix}, \quad \text{and} \quad \mathbf{S} = \begin{bmatrix}\mathbf{S}_1 & \mathbf{S}_2 & \ldots & \mathbf{S}_M\end{bmatrix}.$$

Each row and each column of $\mathbf{Q}$, $\mathbf{R}$ or $\mathbf{S}$ should contain at most one e, while the rest of its elements are equal to ε. Let

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

Define

$$\tilde{\mathbf{A}} = \begin{bmatrix}\mathbf{A}_1 & \varepsilon & & \varepsilon\\ \varepsilon & \mathbf{A}_2 & & \varepsilon\\ & & \ddots & \\ \varepsilon & \varepsilon & & \mathbf{A}_M\end{bmatrix}.$$

Similarly define $\tilde{\mathbf{B}}$, $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{D}}$.

Then we have

$$\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{y}}\end{bmatrix} = \begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}} & \tilde{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{u}}\\ \tilde{\mathbf{v}}\end{bmatrix}. \qquad (4)$$

Outputs of $\mathfrak{M}$ can be expressed as

$$\begin{aligned}\mathbf{x}_{ext} &= \textstyle\bigoplus_{k=1}^{M} (\mathbf{S}_k \mathbf{x}_k) = \mathbf{S}\tilde{\mathbf{x}},\\ \mathbf{y}_{ext} &= \textstyle\bigoplus_{k=1}^{M} \left(\mathbf{Q}_k^T \mathbf{y}_k\right) = \mathbf{Q}^T\tilde{\mathbf{y}}\end{aligned}$$

or, equivalently,

$$\begin{bmatrix}\mathbf{x}_{ext}\\ \mathbf{y}_{ext}\end{bmatrix} = \begin{bmatrix}\mathbf{S} & \varepsilon\\ \varepsilon & \mathbf{Q}^T\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{y}}\end{bmatrix}. \qquad (5)$$

It is assumed that there are no explicitly defined delays associated with transportation of jobs from one system to another - rather these delays can be modeled by an appropriate manufacturing system m_i. Suppose that job $z \in \mathcal{K}_{k,i}$ is routed from m_k to m_i. Job z becomes available to m_i when it is ready to leave m_k. Therefore we have

$$\begin{aligned}\mathbf{u}_i &= \textstyle\bigoplus_{k=1}^{M} (\mathbf{R}_{i,k}\mathbf{x}_k) \oplus \mathbf{Q}_i \mathbf{u}_{ext}\\ &= \begin{bmatrix}\mathbf{R}_{i,1} & \mathbf{R}_{i,2} & \ldots & \mathbf{R}_{i,M}\end{bmatrix}\tilde{\mathbf{x}} \oplus \mathbf{Q}_i\mathbf{u}_{ext}.\end{aligned} \qquad (6)$$

Job z gets removed from m_k when it enters m_i. Therefore we have

$$\begin{aligned}\mathbf{v}_i &= \textstyle\bigoplus_{k=1}^{M} \left(\mathbf{R}_{k,i}^T\mathbf{y}_k\right) \oplus \mathbf{S}_i^T \mathbf{v}_{ext}\\ &= \begin{bmatrix}\mathbf{R}_{1,i}^T & \mathbf{R}_{2,i}^T & \ldots & \mathbf{R}_{M,i}^T\end{bmatrix}\tilde{\mathbf{y}} \oplus \mathbf{S}_i^T\mathbf{v}_{ext}.\end{aligned} \qquad (7)$$

Writing (6) and (7) in terms of $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$ we get

$$\begin{aligned}\tilde{\mathbf{u}} &= \mathbf{R}\tilde{\mathbf{x}} \oplus \mathbf{Q}\mathbf{u}_{ext},\\ \tilde{\mathbf{v}} &= \mathbf{R}^T\tilde{\mathbf{y}} \oplus \mathbf{S}^T\mathbf{v}_{ext}\end{aligned}$$

or, equivalently,

$$\begin{bmatrix}\tilde{\mathbf{u}}\\ \tilde{\mathbf{v}}\end{bmatrix} = \begin{bmatrix}\mathbf{R} & \varepsilon\\ \varepsilon & \mathbf{R}^T\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{y}}\end{bmatrix} \oplus \begin{bmatrix}\mathbf{Q} & \varepsilon\\ \varepsilon & \mathbf{S}^T\end{bmatrix}\begin{bmatrix}\mathbf{u}_{ext}\\ \mathbf{v}_{ext}\end{bmatrix}. \qquad (8)$$

The block diagram illustrating the concept of routing jobs through sub-systems is shown in Figure 3.

Equations (4), (8) and (5) provide composition rules for aggregate system $\mathfrak{M}$. Using these equations one can easily solve for $\begin{bmatrix}\mathbf{x}_{ext}\\ \mathbf{y}_{ext}\end{bmatrix}$ in terms of $\begin{bmatrix}\mathbf{u}_{ext}\\ \mathbf{v}_{ext}\end{bmatrix}$. Substituting (8) into (4) we get

$$\begin{aligned}\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{y}}\end{bmatrix} &= \begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}} & \tilde{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\mathbf{R} & \varepsilon\\ \varepsilon & \mathbf{R}^T\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{y}}\end{bmatrix}\\ &\quad \oplus \begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}} & \tilde{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\mathbf{Q} & \varepsilon\\ \varepsilon & \mathbf{S}^T\end{bmatrix}\begin{bmatrix}\mathbf{u}_{ext}\\ \mathbf{v}_{ext}\end{bmatrix}.\end{aligned}$$

From Theorem 2.1 it follows that

$$\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{y}}\end{bmatrix} = \left(\begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}} & \tilde{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\mathbf{R} & \varepsilon\\ \varepsilon & \mathbf{R}^T\end{bmatrix}\right)^{*} \qquad (9)$$

$$\otimes \begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}} & \tilde{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\mathbf{Q} & \varepsilon\\ \varepsilon & \mathbf{S}^T\end{bmatrix}\begin{bmatrix}\mathbf{u}_{ext}\\ \mathbf{v}_{ext}\end{bmatrix} \qquad (10)$$

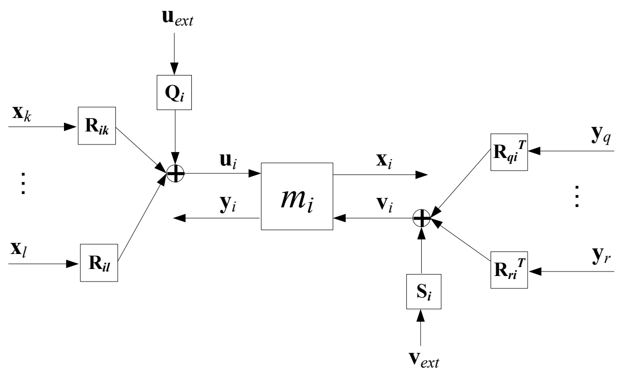

Fig. 3. Composition of manufacturing systems

Substituting (10) into (5) we get

$$\begin{aligned}\begin{bmatrix}\mathbf{x}_{ext}\\ \mathbf{y}_{ext}\end{bmatrix} &= \begin{bmatrix}\mathbf{S} & \varepsilon\\ \varepsilon & \mathbf{Q}^T\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{y}}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{S} & \varepsilon\\ \varepsilon & \mathbf{Q}^T\end{bmatrix}\left(\begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}} & \tilde{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\mathbf{R} & \varepsilon\\ \varepsilon & \mathbf{R}^T\end{bmatrix}\right)^*\\ &\quad \otimes\begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}} & \tilde{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\mathbf{Q} & \varepsilon\\ \varepsilon & \mathbf{S}^T\end{bmatrix}\begin{bmatrix}\mathbf{u}_{ext}\\ \mathbf{v}_{ext}\end{bmatrix}.\end{aligned}$$

Therefore

$$\begin{bmatrix}\mathbf{x}_{ext}\\ \mathbf{y}_{ext}\end{bmatrix} = \mathbf{W}\begin{bmatrix}\mathbf{u}_{ext}\\ \mathbf{v}_{ext}\end{bmatrix}, \tag{11}$$

where $\mathbf{W} =$

$$= \begin{bmatrix}\mathbf{S} & \varepsilon\\ \varepsilon & \mathbf{Q}^T\end{bmatrix}\left(\begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}} & \tilde{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\mathbf{R} & \varepsilon\\ \varepsilon & \mathbf{R}^T\end{bmatrix}\right)^*\begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}} & \tilde{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\mathbf{Q} & \varepsilon\\ \varepsilon & \mathbf{S}^T\end{bmatrix}. \tag{12}$$

Equation (12) gives general expression for the system matrix of $\mathfrak{M}$. This proves that any composition of systems represented by (1) results in a system that is also represented by (1).

Sometimes instead of explicitly specifying $\mathbf{v}_{ext}$ it is assumed that jobs are removed from the system as soon as they are ready to leave the system. In other words machines are never blocked from outside of the system. Then we have

$$\mathbf{v}_{ext} = \mathbf{x}_{ext} = \mathbf{S}\tilde{\mathbf{x}}. \tag{13}$$

Substituting (13) into equations (4), (8) (5) and after some algebraic manipulation it follows that

$$\begin{bmatrix}\mathbf{x}_{ext}\\ \mathbf{y}_{ext}\end{bmatrix} = \mathbf{W}_{nb}\mathbf{u}_{ext} \tag{14}$$

where $\mathbf{W}_{nb} =$

$$\begin{aligned}&= \begin{bmatrix}\mathbf{W}_{nb1}\\ \mathbf{W}_{nb2}\end{bmatrix} = \begin{bmatrix}\mathbf{S} & \varepsilon\\ \varepsilon & \mathbf{Q}^T\end{bmatrix}\left(\begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}} & \tilde{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\mathbf{R} & \varepsilon\\ \mathbf{S}^T\mathbf{S} & \mathbf{R}^T\end{bmatrix}\right)^*\\ &\quad\otimes\begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}} & \tilde{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\mathbf{Q}\\ \varepsilon\end{bmatrix}. \end{aligned}\tag{15}$$

Then $\mathbf{W}_{nb1}\mathbf{u}_{ext}$ gives the times when jobs leave the system and $\mathbf{W}_{nb2}\mathbf{u}_{ext}$ gives the times when they entered the system.

V. DEADLOCK DETECTION

Consider system $\mathfrak{M}$, modeled by matrix $\mathbf{W}$. If the system has deadlocks than some jobs that enter the system will never be able to leave it. Suppose that job $[J^{in}]_k$ is in deadlock and cannot leave the system, then $[\mathbf{x}]_l = +\infty$, where $[J^{out}]_l \equiv [J^{in}]_k$. This means that $\mathbf{W}$ contains elements that are equal to $+\infty$; in other words $\mathbf{W}$ will not exist. On the contrary, if $\mathbf{W}$ exists then the system is free of deadlocks.

Suppose that $\mathfrak{M}$ is a network of subsystems m_i as described in the previous section, such that each m_i is deadlock free. Then from (12) it follows that $\mathbf{W}$ exists (and, therefore, $\mathfrak{M}$ is deadlock free) if and only if

$$\left(\begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}} & \tilde{\mathbf{D}}\end{bmatrix}\begin{bmatrix}\mathbf{R} & \epsilon\\ \epsilon & \mathbf{R}^T\end{bmatrix}\right)^*$$

exists.

VI. MODELS OF MACHINES AND BUFFERS

Next, models of basic nodes in a manufacturing system, such as machines and buffers, are presented.

A. Model of a machine

Consider a machine which processes a set of jobs $\mathcal{J}$. Let J be an ordered set specifying the order in which jobs are processed by the machine. Let $J^{in} = J^{out} = J$. Let $[t]_k$ be the processing time required for job J_k on the machine (no preemption is allowed). It is assumed that the machine is initially empty.

The inputs, $\mathbf{u}$ and $\mathbf{v}$, and the outputs, $\mathbf{x}$ and $\mathbf{y}$, of the system were defined in Section III. Event $[\mathbf{y}]_1$ occurs when event $[\mathbf{u}]_1$ takes pace. For $k > 1$, event $[\mathbf{y}]_k$ occurs immediately after both events $[\mathbf{u}]_k$ and $[\mathbf{v}]_{k-1}$ take place. Therefore,

$$[\mathbf{y}]_k = \begin{cases}[\mathbf{u}]_k \oplus [\mathbf{v}]_{k-1}, & \text{for } k > 1\\ [\mathbf{u}]_1, & \text{for } k = 1.\end{cases} \tag{16}$$

The machine starts processing J_k as early as possible, i.e. at time $[\mathbf{y}]_k$ because the system is uncontrolled. Event $[\mathbf{x}]_k$ occurs when J_k is done processing on the machine. Therefore

$$[\mathbf{x}]_k = [t]_k[\mathbf{y}]_k = [t]_k([\mathbf{u}]_k \oplus [\mathbf{v}]_{k-1}). \tag{17}$$

Writing (16) and (17) in vector form we get

$$\begin{aligned}\mathbf{y} &= \mathbf{u} \oplus \mathbf{H}\mathbf{v},\\ \mathbf{x} &= \mathbf{P}\mathbf{y} = \mathbf{P}(\mathbf{u} \oplus \mathbf{H}\mathbf{v}),\end{aligned}$$

where $\mathbf{H} = \begin{bmatrix}\epsilon & & \epsilon & \epsilon\\ e & & \epsilon & \epsilon\\ & \ddots & & \\ \epsilon & & e & \epsilon\end{bmatrix}$ and $\mathbf{P} = \begin{bmatrix}t_1 & \epsilon & & \epsilon\\ \epsilon & t_2 & & \epsilon\\ & & \ddots & \\ \epsilon & \epsilon & & t_n\end{bmatrix}$. Therefore

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \begin{bmatrix}\mathbf{P} & \mathbf{PH}\\ \mathbf{E} & \mathbf{H}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}. \tag{18}$$

The block diagram representation of a machine is shown in Figure 4.

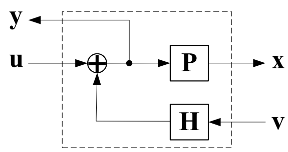

Fig. 4. Model of machine

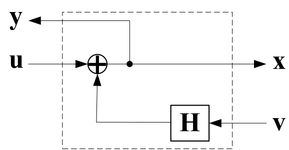

Fig. 5. Model of buffer of unit capacity

B. Model of a unit capacity buffer

McCormick et al. [6] show that a buffer of unit capacity can be represented by a machine having zero processing time for all jobs. Therefore for buffer of unit capacity equation (18) becomes

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \begin{bmatrix}\mathbf{E}\ \mathbf{H}\\ \mathbf{E}\ \mathbf{H}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}, \tag{19}$$

since $\mathbf{P} = \mathbf{E}$. Block diagram representing a buffer of capacity one is shown in Figure 5.

C. Model of an unlimited capacity buffer

Consider a buffer having unlimited storage capacity. Let $J = J^{out} = J^{in}$. J_k enters the buffer as soon as it is available to the buffer, therefore $[\mathbf{y}]_k = [\mathbf{u}]_k$. In addition J_k is ready to leave the buffer immediately after it entered the buffer, therefore $[\mathbf{x}]_k = [\mathbf{y}]_k$. The equations for an infinite buffer are therefore

$$\mathbf{x} = \mathbf{u}$$
$$\mathbf{y} = \mathbf{u},$$

or, in vector form,

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \begin{bmatrix}\mathbf{E} & \epsilon\\ \mathbf{E} & \epsilon\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}.$$

A block diagram representation of infinite buffer is provided in Figure 6.

D. Model of a machine followed by an infinite capacity buffer

A machine followed by an infinite buffer is never blocked because the buffer is able to hold all the parts that exit the machine. Graphically, this model can be represented by connecting a block diagram of machine with a block diagram

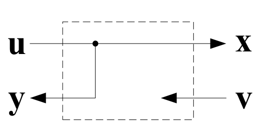

Fig. 6. Model of buffer of infinite capacity

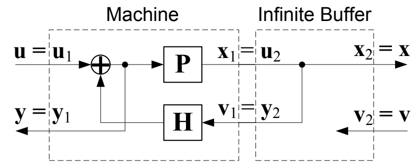

Fig. 7. Model of a machine that is never blocked.

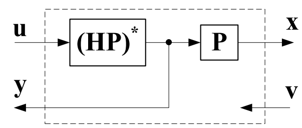

Fig. 8. Alternative block diagram representation of machine that is never blocked.

of infinite buffer as shown in Figure 7. Equations describing the node can be obtained from that block diagram, i.e.

$$\begin{aligned}\mathbf{y} &= \mathbf{u} \oplus \mathbf{H}\mathbf{x},\\ \mathbf{x} &= \mathbf{P}\mathbf{y}.\end{aligned}$$

From which it follows that

$$\begin{aligned}\mathbf{y} &= \mathbf{u} \oplus \mathbf{H}\mathbf{P}\mathbf{y} = (\mathbf{H}\mathbf{P})^{*}\mathbf{u},\\ \mathbf{x} &= \mathbf{P}(\mathbf{H}\mathbf{P})^{*}\mathbf{u}.\end{aligned} \tag{20}$$

Therefore the node is described by

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \begin{bmatrix}\mathbf{P}(\mathbf{H}\mathbf{P})^{*} & \varepsilon\\ (\mathbf{H}\mathbf{P})^{*} & \varepsilon\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}. \tag{21}$$

The outputs of the node do not depend on $\mathbf{v}$ which is due to infinite buffer. Note that $(\mathbf{H}\mathbf{P})^{*}$ always exist because $\mathbf{H}\mathbf{P}$ is a lower triangular matrix. Figure 8 provides an alternative representation of the node which is based on (21).

VII. Example

Consider the following example. A job shop system is represented as a network whose nodes are machines as shown in Figure 9. The numbers above nodes represent the order in which jobs are to be processed on the machines (i.e. 3-4 above m_3 indicates that m_3 will first process job 3 followed by job 4). The labels above arcs indicate routes of jobs through the machines (e.g. jobs 3,4 arrive to machine m_3 from m_2). The numbers inside a node show processing times for all jobs processed by the machine in the order in which operations are performed (e.g. $\{4, 2\}$ inside m_3 means that processing times for jobs 3 and 4 on m_3 are 4 and 2 time units, respectively). It is assumed that there is no intermediate storage between the machines (in general, buffer storage can be modeled by appropriate nodes representing buffers).

Processing time matrices are

$$\mathbf{P}_1 = \begin{bmatrix}1 & \varepsilon\\ \varepsilon & 3\end{bmatrix} \quad \mathbf{P}_2 = \begin{bmatrix}3 & \varepsilon\\ \varepsilon & 5\end{bmatrix}$$
$$\mathbf{P}_3 = \begin{bmatrix}4 & \varepsilon\\ \varepsilon & 2\end{bmatrix} \quad \mathbf{P}_4 = \begin{bmatrix}3 & \varepsilon & \varepsilon\\ \varepsilon & 2 & \varepsilon\\ \varepsilon & \varepsilon & 1\end{bmatrix}.$$

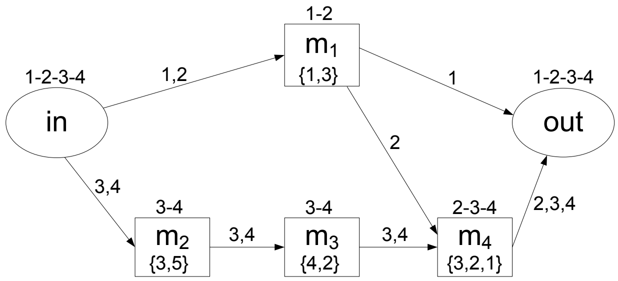

Fig. 9. Sample manufacturing layout

The routing of jobs is represented by routing matrices $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{S}$, whose values can be obtained directly from the manufacturing layout in Figure 9. For example, routing of jobs from m_1 to m_4 is given by

$$\mathbf{R}_{4,1} = \begin{bmatrix} \varepsilon & e \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix}.$$

Job 2 is routed directly from m_1 to m_4, it is the 1-st job to be processed on m_4 and is the 2-nd job to be processed on m_1. Therefore, $[\mathbf{R}_{4,1}]_{1,2} = e$.

Each machine m_i is modeled by an equation of the form (18). For m_1 we have

$$\begin{aligned} \mathbf{W}_1 &= \begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 \\ \mathbf{C}_1 & \mathbf{D}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_1\mathbf{H} \\ \mathbf{E} & \mathbf{H} \end{bmatrix} \\ &= \begin{bmatrix} \begin{bmatrix} 1 & \varepsilon \\ \varepsilon & 3 \\ e & \varepsilon \\ \varepsilon & e \end{bmatrix} & \begin{bmatrix} \varepsilon & \varepsilon \\ 3 & \varepsilon \\ \varepsilon & \varepsilon \\ e & \varepsilon \end{bmatrix} \end{bmatrix}. \end{aligned}$$

In a similar way, we find $\mathbf{W}_2$, $\mathbf{W}_3$ and $\mathbf{W}_4$, which allows us to obtain $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{D}}$. Assuming that jobs are removed from the system as soon as they are ready to leave, it follows from (15) that

$$\begin{bmatrix} \mathbf{x}_{ext} \\ \mathbf{y}_{ext} \end{bmatrix} = \begin{bmatrix} 1 & \varepsilon & \varepsilon & \varepsilon \\ 7 & 6 & 6 & 6 \\ 9 & 8 & 9 & 8 \\ 10 & 9 & 11 & 9 \\ e & \varepsilon & \varepsilon & \varepsilon \\ 1 & e & e & e \\ \varepsilon & \varepsilon & e & \varepsilon \\ e & e & 3 & e \end{bmatrix} \mathbf{u}_{ext} \qquad (22)$$

Suppose that all the jobs are made available to the system at time zero, which means that $\mathbf{u}_{ext} = \begin{bmatrix} e & e & e & e \end{bmatrix}^T$. It follows from (22) that the times when the jobs actually enter the system are given by $\mathbf{y}_{ext} = \begin{bmatrix} e & 1 & e & 3 \end{bmatrix}^T$ and the times when the jobs are ready to leave the system are given by $\mathbf{x}_{ext} = \begin{bmatrix} 1 & 7 & 9 & 11 \end{bmatrix}^T$. Job 4 is the last job to leave the system (at time 11), therefore the system's makespan is 11 time units.

VIII. CONCLUSION

Described in this paper is the new approach to model deterministic manufacturing systems. The approach is based on block diagrams. 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 models of basic modeling blocks of manufacturing systems such as machines and buffers were developed.

The approach can be readily implemented in computer software as it basically involves operations performed on 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, work in process, machine utilization, etc.

The modeling approach can be used in manufacturing system design and scheduling applications. The approach is not limited to deterministic manufacturing systems; for example, with minor modifications it can be applied to modeling parallel computer processing networks.

REFERENCES

[1] D. Findeisen. *System Dynamics and Mechanical Vibrations: An Introduction*. Springer, 2000.
[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] Tae-Eog Lee and M.E. Posner. Performance measures and schedules in periodic job shops. *Operations Research*, 45(1):72–90, 1997.
[9] 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.
[10] 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.
[11] 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.
[12] B. Heidergott. A characterisation of (max,+)-linear queueing systems. *Queueing Systems*, 35(1–4):237–262, 2000.
[13] 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.
[14] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity*. John Wiley and Sons, West Sussex, England, 1992.
[15] 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.
[16] N. Krivulin. The max-plus algebra approach in modelling of queueing networks. In *Summer Computer Simulation Conference*, pages 485–490, Portland, OR, July 1996.
[17] 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.
[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.