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

Matrix-Based Approach to Sequence analysis in Multiple Reentrant Flowlines

Tamara Petrovic, Stjepan Bogdan
LARICS - Laboratory for Advanced Robotics and Intelligent Control Systems
Faculty of Electrical Engeneering and Computing
University of Zagreb
http://flrcg.rasip.fer.hr

***Abstract*— This paper presents a method for extension of matrix model of manufacturing system in order to provide an efficient tool for analysis of systems with various dispatching sequences of shared resources. Proposed method is used for transformation of system matrices in linear max-plus model. Once the linear model is determined sequence feasibility can be checked. Furthermore, the method provides a straightforward procedure for production cycle calculation and resource utilisation. Efficiency of presented technique is demonstrated on a manufacturing system example at the end of the paper.**

I. Introduction

There are many approaches to modeling, simulation and control design for manufacturing systems (MS), including automata [1], Petri Nets [2], alphabet-based approaches, perturbation methods, expert system design, and so on. In this paper a matrix based model of flexible manufacturing system (FMS) that is a part of a detailed design of manufacturing systems is used [3]. This matrix framework is very convenient for computer simulation, as well as for supervisory control design [4], [5]. It is straightforward to write down the matrix description for a specific manufacturing system, since the matrices of the model are given by the bill of material [6], Steward's sequencing matrix [7], the resource requirements matrix, assembly trees and existing dispatching algorithms. In addition, the matrix based formulation can be easily modified if there are changes in product requirements or resources available, making workcell control more flexible and reconfigurable.

In the paper the following three assumptions that define the sort of discrete-part manufacturing systems are made:

No pre-emption - once assigned, a resource cannot be removed from a job until it's completed,

Mutual exclusion - a single resource can be used for only one job at a time,

Hold while waiting - a process holds the resources already allocated to it until it has all the resources required to perform a job.

In addition to these assumptions, it is assumed that there are no machine failures.

When a multitude of jobs requesting the same shared resource are simultaneously activated, a *conflict* is said to have occurred and a decision is needed as to which job the resource should be allocated to. This type of priority assignment in resource allocation constitutes the *problem of dispatching*. In the paper this problem is solved by introducing the repeatable sequence of operations of shared resource, called *dispatching sequence*, that is given and fixed during the operating time of the manufacturing system. Performance analysis based on matrix model is done by using simulation. In order to provide a more efficient method for analysis of dispatching sequence of shared resources, an extension of the matrix model is proposed. The proposed method transforms the matrix model in max-plus framework, and thereby provides determination of a) feasibility of particular sequence and b) production cycle and resource utilisation.

This paper is organised in the following way: first the system matrices and equations in *and/or* algebra that fully describe an MS are introduced. In order to be able to investigate dynamic phenomena in an MS, time is included into the matrix model. Then the procedure for matrix model to max-plus model transformation is described, followed by an example of sequence analysis based on the proposed method.

II. Discrete event systems model in matrix form

Given a set of jobs and a set of resources that compose a manufacturing system, the system activities can be presented in the form of IF-THEN rules. Each rule corresponds to a component of the *logical state vector*, denoted $\mathbf{x}$. A job is said to be *activated (started)* when all the preconditions (IF part) for its execution are satisfied.

All matrix operations are defined to be in *and/or* algebra, denoted Δ and ∇, where standard multiplication is replaced by logical *and*, and standard addition by logical *or*. Given a natural number vector $\mathbf{a} = \begin{bmatrix} a_1 \ a_2 \ \ldots \ a_n \end{bmatrix}^T$, its negation $\bar{\mathbf{a}} = \begin{bmatrix} \bar{a}_1 \ \bar{a}_2 \ \ldots \ \bar{a}_n \end{bmatrix}^T$ is such that $\bar{a}_i = 0$ if $\bar{a}_i > 0$ and 1 otherwise.

A. System matrices

A system is fully described with the following matrices:

$\mathbf{F}_v$ is *a job sequencing matrix:* $\mathbf{F}_v(i, j) = 1$ if job j contributes to construction of the i-th component of the logical state vector, otherwise $\mathbf{F}_v(i, j) = 0$;

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

$\mathbf{F}_r$ is *a resource-requirements matrix:* $\mathbf{F}_r(i,j) = 1$ if resource j contributes to construction of the i-th component of the logical state vector, otherwise $\mathbf{F}_r(i,j) = 0$;

$\mathbf{F}_u$ is *an input matrix:* $\mathbf{F}_u(i,j) = 1$ if an input (raw parts entering the system) j contributes to the construction of the i-th component of the logical state vector, otherwise $\mathbf{F}_u(i,j) = 0$

$\mathbf{S}_v$ is *a job-start matrix*: $\mathbf{S}_v(i,j) = 1$ if the j-th component of the logical state vector is a prerequisite to start job i, otherwise $\mathbf{S}_v(i,j) = 0$.

$\mathbf{S}_r$ is *a resource-release matrix*: $\mathbf{S}_r(i,j) = 1$ if the j-th component of the logical state vector is a prerequisite to start the release of resource i, otherwise $\mathbf{S}_r(i,j) = 0$. It should be noted that for a shared resource at least two rules exist that release it.

$\mathbf{S}_v$ is *an output matrix:* $\mathbf{S}_v(i,j) = 1$if j-th component of the logical state vector is a prerequisite for output i. Otherwise $\mathbf{S}_v(i,j) = 0$.

B. Recursive matrix Model

Generally, the complete task plan could be given by the system matrices $\mathbf{F}_v, \mathbf{S}_v, \mathbf{F}_r, \mathbf{S}_r$, defined above. Denoting the discrete event iteration number with k, the logical state vector is calculated each time an event takes place, i.e. a job is completed, resource becomes idle or part enters the system:

$$\bar{\mathbf{x}}(k) = \mathbf{F}_v \Delta \bar{\mathbf{v}}_c(k-1) \nabla \mathbf{F}_r \Delta \bar{\mathbf{r}}_c(k-1) \nabla \mathbf{F}_u \Delta \bar{\mathbf{u}}(k-1), \quad (1)$$

where $\mathbf{v}_c$ is *job completed vector*, $\mathbf{r}_c$ is *idle resource vector* and $\mathbf{u}$ is *input vector* (representing raw parts entering the cell).

The *system vector* $\mathbf{m}(k)$ is introduced as:

$$\mathbf{m}(k) = \left[\mathbf{u}(k)\ \mathbf{v_c}(k)\ \mathbf{r_c}(k)\ \mathbf{y}(k)\right], \quad (2)$$

where $\mathbf{y}$ is *output vector* (representing parts leaving the cell).

Then, the recursive matrix model can be written in the following form:

$$\begin{aligned} \bar{\mathbf{x}}(k) &= \mathbf{F} \Delta \bar{\mathbf{m}}(k-1), \qquad \mathbf{m}(0) = \mathbf{m}_0 \\ \mathbf{m}(k) &= \mathbf{m}(k-1) + \left[\mathbf{S} - \mathbf{F}^T\right] \mathbf{x}(k) \end{aligned} \quad (3)$$

with

$$\mathbf{S} = \begin{bmatrix}\mathbf{S}_u^T & \mathbf{S}_v^T & \mathbf{S}_r^T & \mathbf{S}_y^T\end{bmatrix}^T,$$

$$\mathbf{F} = \begin{bmatrix}\mathbf{F}_u^T & \mathbf{F}_v^T & \mathbf{F}_r^T & \mathbf{F}_y^T\end{bmatrix},$$

where $\mathbf{S_u} = [\mathbf{0}]$, $\mathbf{F_y} = [\mathbf{0}]$ are null-matrices required for keeping matrix dimensions consistent.

Hybrid matrix model (3) does not capture the system dynamics. By tracking $\mathbf{m}(k)$ only logical activities of the system are reconstructed. To make the performance analysis of the system possible, the operational times should be incorporated in model (3).

C. Modeling System Dynamics

To keep track of job time duration, the system dynamics is incorporated into the matrix model in the form of a *lifetime* [8], [9]. An integer number n_i, called a lifetime, is associated with each task i in an MS. A *shift (delay) operator* q is introduced: $y(q) = q^{-n}x(q)$ corresponds to $y(k) = x(k-n)$, i.e. y is delayed n sampling intervals after x.

The dynamic matrix model of an MS is obtained by including the shift operator q and operations lifetimes in recursive matrix model (3):

$$\begin{aligned} \bar{\mathbf{x}}(q) &= \mathbf{F} \Delta q^{-1} \bar{\mathbf{m}}(q), \qquad \mathbf{m}(0) = \mathbf{m}_0 \\ \mathbf{m}(q) &= q^{-1}\mathbf{m}(q) + \left[\mathbf{T}(q) - \mathbf{F}^T\right] \mathbf{x}(q) \end{aligned} \quad (4)$$

where

$$\mathbf{T}(q) = \begin{bmatrix}\mathbf{S}_u^T & \mathbf{T}_v^T(q) & \mathbf{T}_r^T(q) & \mathbf{S}_y^T\end{bmatrix}^T$$

$\mathbf{T_v}$ and $\mathbf{T}_r$ are operation and resource release *delay matrices* with elements representing operations lifetimes. Delay matrices are obtained by replacing each entry "1" in $\mathbf{S_v}$ and $\mathbf{S_r}$ with a shift operand representation of the corresponding lifetime.

D. Max-Plus model

The implicit max-plus model of the system is given with:

$$\mathbf{x}(k) = \mathbf{A}_0 \otimes \mathbf{x}(k) \oplus \mathbf{A}_1 \otimes \mathbf{x}(k-1), \quad (5)$$

where $\mathbf{x}(k)$ denotes k-th occurrence of events included in vector $\mathbf{x}$. Symbols $\otimes$ and $\oplus$ stand for max-plus multiplication and addition, respectively. Once matrices $\mathbf{A}_0$ and $\mathbf{A}_1$ are know, system matrix $\mathbf{A}$ of an explicit max-plus model $\mathbf{x}(k) = \mathbf{A} \otimes \mathbf{x}(k-1)$ is obtained as $\mathbf{A} = \left(\bigotimes_{i=0}^{n} \mathbf{A}_0^i\right) \otimes \mathbf{A}_1$. Once the explicit system model is determined, production cycle, resources utilisation and other system performance properties can be easily calculated [3].

E. Deriving max-plus framework from matrix model

Determining the max-plus model of the manufacturing system with no shared resources that is initially given in matrix form is straightforward, i.e.

$$\mathbf{x}(k) = \mathbf{D}_v \otimes \mathbf{x}(k) \oplus \mathbf{D}_r \otimes \mathbf{x}(k), \quad (6)$$

where $\mathbf{D}_v$ is obtained by multiplication $\mathbf{F}_v\mathbf{T}_v$ where q^{-n} is replaced with n, and 0 with $\varepsilon = -\infty$. Matrix $\mathbf{D}_r$ is determined in the same way by multiplication $\mathbf{F}_v\mathbf{T}_v$.Vector $\mathbf{x}(k)$ in (6) includes both implicit and explicit relations between system events.

III. SEQUENCING IN MATRIX FORM

Since the max-plus representation is feasible only for decision-free discrete event systems (event graphs), for systems with shared resources, a control strategy that provides conflict-free dispatching should be determined prior to transformation of the matrix model to max-plus. In the matrix formulation, the closed loop system including both the workcell and the controller has the following form:

$$\bar{\mathbf{x}}(q) = \mathbf{F}\Delta q^{-1}\bar{\mathbf{m}}(q)\nabla\mathbf{F}_d\Delta\bar{\mathbf{u}}_d(q), \qquad (7)$$

where $\mathbf{F}_d$ is *the dispatching matrix*, and $\mathbf{u}_d$ is *the dispatching vector*. The structure and the value of dispatching vector depend on applied dispatching policy. Generally, vector $\mathbf{u}_d$ is determined as a function of feedback signals comprised in vector $\mathbf{m}$.

For a particular form of $\mathbf{F}_d$ and $\mathbf{u}_d$, the sequence of operations, executed by shared resource repeats, hence, the manufacturing system demonstrates cyclic behaviour. The question is how efficient is the resources utilisation for different sequences of operations? Furthermore, is a particular sequence, chosen by the production manager, feasible at all?

To point out this subject more clearly let us examine simple workcell shown in Fig.1. The cell consist of two machines - A and B and one robot - R. Two types of parts, *a* and *b*, are processed in two flowlines in the following way. Both parts are brought into the cell by input conveyers. Entering the cell, part *a* is picked up by the robot (operation RP1) and transported to the machine A. When processing in the machine A is finished, the robot removes the part from the machine and leaves it on the output conveyer (operation RP2). Upon arrival, part *b* is processed in the machine B and then taken by the robot to its output conveyer (operation RP3).

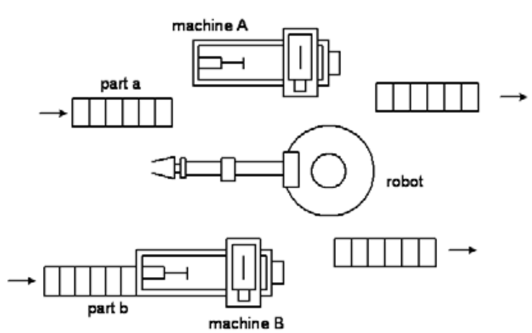

Fig. 1. An example of simple workcell

Evidently, the robot is a shared resource that executes three tasks: RP1, RP2 and RP3. Let machine B operational time be much larger than operational time of machine A. Then, regarding resource utilisation and production cycle, the system performances would be better when more parts *a* then parts *b* are produced in one production cycle. When feasibility of sequences in the given workcell is analysed, it is apparent that sequence {RP1, RP1, RP2, RP3} is not feasible since its execution leads the system into deadlock caused by the overload of parts in machine A.

Although derived conclusions are obvious due to the fact that the workcell in example comprises only three resources, in case of complex manufacturing systems with tens of various resources, these phenomena are observable only through a firm mathematical framework.

Dispatching policy defined as a sequence of operations for each shared resource. In the paper, two types of sequences are distinguished: *sequences with no repetition* (where each operation appears in a defined sequence exactly once) and *sequences with repetition*. Methods for deriving max-plus models for these two types of sequences differ and are described in the text that follows.

A. Sequence with no repetition

Sequence matrix $\mathbf{\Phi}$ is defined as a binary square matrix with element $\Phi(i,j) = 1$ if shared resource operation v_j follows immediately after shared resource operation v_i, otherwise $\Phi(i,j) = 0$. Matrix $\mathbf{\Phi}$ is a square matrix whose dimensions equal the number of operations of a shared resource. For the system in Fig.1. and sequence {RP1, RP2, RP3} the sequence matrix is:

$$\mathbf{\Phi} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

To obtain an implicit max-plus model in form of (5), the sequence matrix is first split into two parts, $\mathbf{\Phi}_0$ and $\mathbf{\Phi}_1$, with $\mathbf{\Phi}_0$ describing inner part of the sequence, and $\mathbf{\Phi}_1$ connecting the last and the first operation of the sequence ($\mathbf{\Phi}_1(i,j) = 1$ if i is the last and j the first operation in the sequence). For sequence in the example one gets:

$$\mathbf{\Phi}_0 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{\Phi}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Matrices $\mathbf{A}_0$ and $\mathbf{A}_1$ are obtained from system matrices as:

$$\begin{aligned} \mathbf{A}_0 &= \mathbf{D}_v \oplus \left[\mathbf{D}_r \odot \left(\mathbf{F}_d \cdot \left(\mathbf{\Phi}_0 \cdot \mathbf{F}_{dr}^T\right)\right)\right] \\ \mathbf{A}_1 &= \mathbf{D}_r \odot \left(\overline{\mathbf{x}_d \cdot \mathbf{x}_{dr}^T} + \mathbf{F}_d \cdot \left(\mathbf{\Phi}_1 \cdot \mathbf{F}_{dr}^T\right)\right), \end{aligned} \qquad (8)$$

where $\odot$ denotes matrix element by element standard multiplication. Vector $\mathbf{r}_{sh}$ is a shared resource vector, $r_{sh}(i) = 1$ if resource i is a shared resource. In the example resource vector $\mathbf{r} = \begin{bmatrix} \mathrm{A} & \mathrm{R} & \mathrm{B} \end{bmatrix}^T$ and $\mathbf{r}_{sh} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$. Vector $\mathbf{x}_{dr}$ and $\mathbf{x}_d$ are determined according to: $\mathbf{x}_{dr} = \mathbf{S}_r^T \Delta \mathbf{r}_{sh}$ and $\mathbf{x}_d = \mathbf{F}_r \Delta \mathbf{r}_{sh}$. Elements of matrix $\mathbf{F}_{dr}$ ($\mathbf{F}_d$) are calculated from $\mathbf{x}_{dr}$ ($\mathbf{x}_d$) as:

$$f_{dr}(i,j) = \begin{cases} 1 & \text{if } x_{dr} = 1 \text{ and } j = \sum_{k=1}^{i} x_{dr}(k) \\ 0 & \text{otherwise} \end{cases} \qquad (9)$$

Matrix $\mathbf{A_0}$, determined according to (8), describes implicit relations between system events, i.e. duration of all operations and resource releases in inner part of the sequence.

Matrix $\mathbf{A_1}$, determined according to (8), includes explicit relations between system events, i.e. releases of non-shared resources and release of shared resource after the last operation in the sequence.

Feasibility of predetermined sequence can be checked by system matrix $\mathbf{A}_0$. In case:

$$\left(\mathbf{A}_0 \oplus \mathbf{A}_0^2 \oplus \ldots \oplus \mathbf{A}_0^n\right)_{ii} \neq \varepsilon, \forall i, \tag{10}$$

the sequence is feasible. Matrix $\mathbf{A_0}$ represents places in event graph with no initial marking. If there is a circle of places with no markings in the initial state then the system will end up in deadlock [10].

B. Sequence with repetition

To be able to use the previously given procedure for determining the max-plus model in form of (6), the matrix model of the system should be modified. First, each flowline in the system is described with a separate matrix model. For the system in Fig.1., that consists of two flowlines, one has (matrices $\mathbf{S_v}$ and $\mathbf{S_r}$ are omitted):

1) part *a* production line

$$\mathbf{F}_{v1} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{F}_{r1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{v1}(q) = \begin{bmatrix} q^{-n_1} & 0 & 0 & 0 \\ 0 & q^{-n_2} & 0 & 0 \\ 0 & 0 & q^{-n_3} & 0 \end{bmatrix}$$

$$\mathbf{T}_{r1}(q) = \begin{bmatrix} 0 & 0 & q^{-r_2} & 0 \\ 0 & q^{-r_1} & 0 & q^{-r_3} \end{bmatrix}$$

2) part *b* production line

$$\mathbf{F}_{v2} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{F}_{r2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{v2}(q) = \begin{bmatrix} q^{-n_4} & 0 & 0 \\ 0 & q^{-n_5} & 0 \end{bmatrix}$$

$$\mathbf{T}_{r2}(q) = \begin{bmatrix} 0 & q^{-r_4} & 0 \\ 0 & 0 & q^{-r_5} \end{bmatrix}$$

Flowline subsystems matrices $\mathbf{F}_{vi}, \mathbf{F}_{ri}, \mathbf{S}_{vi}, \mathbf{S}_{ri}, \mathbf{T}_{vi}, \mathbf{T}_{ri}$ are used to form *the extended matrix model* of the system. The number of times each flowline is added in extended matrix model equals the number of repetitions of a particular flowline in predefined sequence. For system shown in Fig.1. sequence with repetition is, for example, $\{$ RP1, RP2, RP1, RP2, RP3 $\}$. In given sequence, part *a* flowline appears twice and part *b* flowline once, hence, part *a* flowline is added in extended matrix model twice and part *b* flowline once.

Elements of extended matrix model matrices are determined as follows. Let $\mathbf{F}_v^k$ be m_1 x n_1 extended system matrix obtained by adding together k flowlines, and let $\mathbf{F}_{vj}$ be m_2 x n_2 flowline matrix that should be added to the extended system matrix as $(k+1)$-st. Then, matrix $\mathbf{F}_v^{k+1}$ is calculated as:

$$f_v^{k+1}(i,j) = \begin{cases} f_v^k(i,j) & \text{for } i \leq m_1 \text{ and } j \leq n_1 \\ f_{vj}(i-m_1, j-n_1) & \text{for } m_1 < i \leq m_1 + m_2 \\ & \text{and } n_1 < j \leq n_1 + n_2 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The same principle is used for calculation of matrices $\mathbf{S}_v$ and $\mathbf{T}_v$.

Let $\mathbf{F}_r^k$ be m_1 x n_1 extended system matrix obtained by adding together k flowlines, and let $\mathbf{F}_{rj}$ be m_2 x n_2 flowline matrix that should be added to the extended system matrix as $(k+1)$-st. Then, matrix $\mathbf{F}_r^{k+1}$ is calculated as:

$$f_r^{k+1}(i,j) = \begin{cases} f_r^k(i,j) & \text{for } i \leq m_1 \text{ and } j \leq n_1 \\ f_{rj}(i-m_1, j-n_1) & \text{for } m_1 < i \leq m_1 + m_2 \\ & \text{and } n_1 < j \leq n_1 + n_2 \\ f_r^k(i,j) & \text{if } l \text{ and } j \text{ are} \\ & \text{the same resource} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Transpose of matrices $\mathbf{S}_r$ and $\mathbf{T}_r$ can be calculated by using (12) as well.

Once extended system matrices are determined, the procedure for deriving the max-plus model is the same as for sequences with no repetition, having in mind that more then one operation in extended model belongs to one operation in the original model. The procedure for determining the feasibility of the sequence is the same as well and given with (10).

IV. Implementation results

The proposed sequence analysis method will be validated on the system shown in Fig.2.

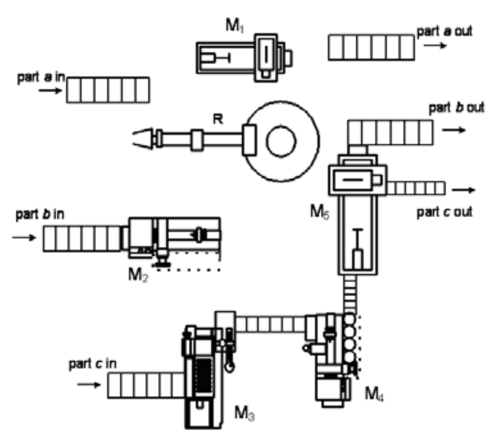

Fig. 2. An example of simple workcell with three flowlines

The system contains three flowlines with two shared resources, robot R, working on parts *a* and *b*, and machine M5, having tasks on parts *b* and *c*. Parts visit resources in the following order: part a : R(a_1)-M1-R(a_2), part b : M2-R-M5, part c : M3-M4-M5. Three different sequences for each shared resource have been analysed:

1) Seq. 1 robot R $\Rightarrow \{a_1\ ,\ a_2,\ a_3\}$
   machine M5 $\Rightarrow \{b,\ c\}$
2) Seq. 2 robot R $\Rightarrow \{a_1,\ a_2,\ a_1,\ a_2,\ a_1,\ a_2,\ b\}$
   machine M5 $\Rightarrow \{b,\ c,\ c\}$
3) Seq. 3 robot R $\Rightarrow \{a_1,\ a_2,\ a_1,\ a_2,\ a_1,\ a_2,\ a_1,\ a_2,\ b\}$
   machine M5 $\Rightarrow \{b,\ c,\ c,\ c\}$

Operations lifetimes have been the same for all three sequences. Fig. 3. shows the number of produced parts per system cycle for predetermined sequences. It can be seen that machine M2 is the system bottleneck, since the shortest flowline cycles are obtained in case of sequence 3, i.e. in that case 5 parts *a* and 3 parts *c* have been processed while only 1 part *b* left the system during one cycle.

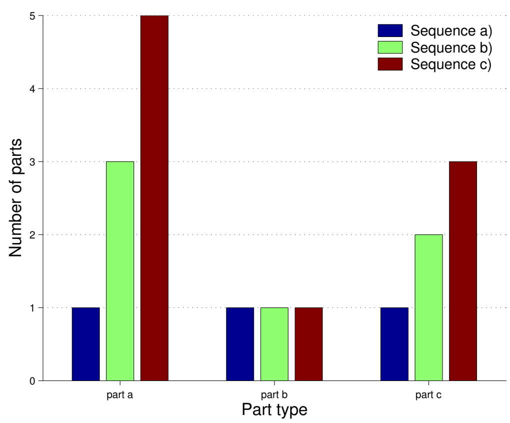

Fig. 3. Number of produced parts per system cycle for predetermined sequences

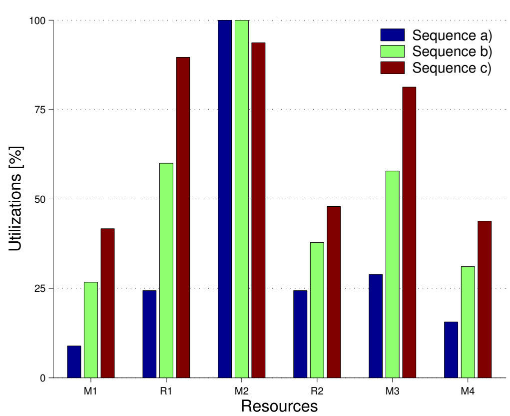

Fig. 4. Resource utilisation for predetermined sequences

Resource utilisation for all three sequences is shown in Fig.4. These results clearly show improvement in work-in-progress in case of sequence 3 since average resources utilisation increased from 40.4 % for sequence 1 to 79.6 % for sequence 3.

V. Conclusions

In this paper an extension of the matrix model is proposed, in order to provide a method for analysing dispatching sequences of shared resources. As we demonstrate, the proposed method transforms matrix models in max-plus framework, thus providing determination of a) feasibility of particular sequence, and b) flowline production cycle and resource utilisation.

Based on sequence analysis one is able to decide which one of predetermined sequences for shared resource scheduling is optimal in the sense of work-in-progress.

Results presented herein are the first step in the development of a dispatching controller synthesis technique that should provide mechanisms for optimal sequence determination (in the sense of resources utilisation) once structural properties of manufacturing system are given in matrix form.

References

[1] W. M. Wonham, "Supervisory control of interacting discrete event systems," 2005.
[2] J. M. Proth and X. Xie, *Petri nets: a Tool for Design and Management of manufacturing systems*. Wiley, 1996.
[3] S. Bogdan, F. Lewis, Z. Kovacic, and J. Mireles, *Manufacturing systems control design: A matrix based approach*. Springer, 2006.
[4] D. A. Tacconi and F. L. Lewis, "A new matrix model for discrete event systems: application to simulation," *IEEE Control Systems Magazine*, vol. 17, pp. 62–71, Oct 1997.
[5] F. L. Lewis and H. H. Huang, "A new matrix model for discrete event systems," *Flexible manufacturing Systems: Recent Developments*, 1994.
[6] H. Noori and R. Radford, *Production and Operations Management*. New York: McGraw-Hill, 1995.
[7] D. V. Steward, *Systems Analysis and Management: Structure, Strategy and Design*. Petrocelli books, 1981.
[8] N. Smolic-Rocak, S. Bogdan, Z. Kovacic, T. Reichenbach, and B. Birgmajer, "Dynamic modeling anad simulation of fms by using vrml," in *CD Proceedings of 15th IFAC World Congress*, 2002.
[9] S. Bogdan, Z.Kovacic, and N. Smolic-Rocak, "Modeling and simulation of manufacturing systems," in *CD Proceedings of EUROSIM*, 2007.
[10] G. Cohen, S. Gaubert, and J. P. Quadrat, "Max-plus algebra and system theory: where we are and where to go now," *Annual reviews in control*, vol. 23, pp. 207–219, 1999.