Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Dealing with Mutual Exclusion Sections in Production Systems: from Shared Resources to Parallel TEG's ⋆

Olivier Boutin *,** **Bertrand Cottenceau** ** **Anne L'Anton** *

* *Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), UMR CNRS 6597*
1 rue de la Noë – BP 92101 – 44321 Nantes Cedex 03, France
e-mail: {olivier.boutin, anne.l-anton}@irccyn.ec-nantes.fr
** *Laboratoire d'Ingénierie des Systèmes Automatisés (LISA), EA 4014*
62 avenue Notre Dame du Lac – 49000 Angers, France
e-mail: bertrand.cottenceau@istia.univ-angers.fr

Abstract: This paper deals with the disambiguation of the behaviour of Petri nets including shared resources. In the production management context, they are often used for the modelling of manufacturing cells. But this representation has a poor transposition into dioid algebra.
In this article, we design a method to describe such a phenomenon in a dioid of interval. The latter expresses this class of Petri nets models in a formal way. Their input/output behaviours are guaranteed to be greater than the lower bound of the reference model set and lower than the upper bound of this set. In fact, the resource sharing problem is turned into a time uncertainty problem, concerning the access to the shared resource. In this new problem, time uncertainties are bounded and can be described by intervals.
Both bounds "confining" the behaviours of the studied production systems in intervals can be manipulated in the scope of the $\overline{\mathbb{Z}}_{max}$ algebra, even though the original systems are not $\overline{\mathbb{Z}}_{max}$ linear by essence.

Keywords: Discrete event systems; Modelling; Simulation; Petri nets; Manufacturing plant operations; Dioid algebra; Interval analysis.

1. INTRODUCTION

Discrete Event Systems (DES) span from transportation, communication or computer networks to manufacturing systems. Many optimization problems are *nonlinear* in traditional arithmetic but appear to be linear over dioids (Baccelli et al. [1992], Heidergott et al. [2005]). Particularly, some linear state representations in dioids can describe the behaviour of Timed Event Graphs (TEG) (Cohen et al. [1989]).

In the specific production management context, TEG's appropriately model manufacturing phenomena such as delays and synchronizations (Trouillet and Benasser [2002], Amari et al. [2004]). However, *shared resources* phenomena cannot be represented in TEG's, because a place cannot have more than one incoming or outgoing arc in that kind of diagram. In job-shops, decisions have to be taken, for a given piece of material to go on one path or another. In practice, the junction in the material paths can be considered as a shared resource since a mutual exclusion policy is usually applied to such portions. That amounts to saying that "usual" dioids can only be used when studying flow-shops. (Boutin et al. [2007]).
Other models such as automata (for instance $\overline{\mathbb{Z}}_{max}$ automata (Gaubert [1995], Al Saba et al. [2006])) are usually better suited to study those systems. Nevertheless, they require a cyclic assignment policy, which is not the case in our study cases. We have not found in literature any other work showing a linear modelling in a dioid algebra of shared resources in production systems.

In this paper we present a shared resource assignment policy such that the behaviour of the system could be described by intervals of time uncertainties. It is a compromise between a cyclic assignment of the resource to the competing processes and a first in first out queuing policy. For this policy, we show that a problem of a shared resource between two sub-systems can be turned into a problem with uncertain delays and unsynchronized TEG's, by means of time uncertainties on the access to the resource and on its unavailability durations. However, let us note that there is no equivalence between the two representations.

The results of this study provide a method to formally represent job-shops (and not only flow-shops) in the dioid of intervals introduced in Lhommeau et al. [2004]. TEG's with uncertain delays can be transposed as *linear* equations over this dioid. This should lead to an alternative to simulation methods for the study of job-shops, and to the automation of the synthesis for job-shop controllers for just-in-time behaviour.

⋆ Many thanks to Didier Lime (from IRCCyN MOVES team) and Jean-Jacques Loiseau (from IRCCyN ACSED team) for their support on Petri nets and formal proofs respectively

In the following section, we introduce theoretical aspects of linear systems, such as dioids and the corresponding graphical formalism, namely the TEG's. Afterwards, we describe how we deal with mutual exclusion and the class of systems we focus on. And then in section 4 we illustrate our method on an application case based on an eight shaped production system.

2. LINEAR SYSTEMS

2.1 Elements of the dioid theory

We recall in this section some aspects of the dioid theory. The reader is invited to consult Baccelli et al. [1992] or Cohen et al. [1989] for an exhaustive presentation.

Definition 1. (Dioid). A dioid is a set $\mathcal{D}$ endowed with two inner operations denoted $\oplus$ (sum) and $\otimes$ (product) [1]. They are both associative (i.e. $\forall (a,b,c) \in \mathcal{D}^3, (a \star b) \star c = a \star (b \star c)$, here and below, the symbol $\star$ denotes any of the two operations $\oplus$ and $\otimes$) and admit neutral elements denoted ε and e respectively. The sum is also commutative (i.e. $\forall (a,b) \in \mathcal{D}^2, a \oplus b = b \oplus a$) and idempotent (i.e. $\forall a \in \mathcal{D}, a \oplus a = a$). The product distributes over the sum (i.e. $\forall a,b,c \in \mathcal{D}^3, a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ and $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$) and the element ε is absorbing for the product (i.e. $\forall a \in \mathcal{D}, \varepsilon \otimes a = a \otimes \varepsilon = \varepsilon$).

Definition 2. (Order relation). An order relation $\succeq$ can be associated with a dioid $\mathcal{D}$ by the following equivalence:
$\forall (a,b) \in \mathcal{D}^2, a \succeq b \Leftrightarrow a = a \oplus b$.
The operations $\oplus$ and $\otimes$ are consistent with the order $\succeq$ in the following sense:
$\forall (a,b,c) \in \mathcal{D}^3$, if $a \succeq b$, then $a \star c \succeq b \star c$ and $c \star a \succeq c \star b$.

Definition 3. (Completeness). A dioid $\mathcal{D}$ is said to be complete if it is closed for infinite sums and if the product distributes over infinite sums too, i.e. if $\forall a \in \mathcal{D}$ and $\forall \mathcal{A} \subseteq \mathcal{D}, a \otimes (\bigoplus_{x \in \mathcal{A}} x) = \bigoplus_{x \in \mathcal{A}} a \otimes x$.

Example 4. ($\overline{\mathbb{Z}}_{max}$ and $\overline{\mathbb{Z}}_{min}$ dioids). The set $\overline{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty, +\infty\}$ endowed with the *maximum* operator as sum and the *classical sum* $+$ as product is a complete dioid, usually noted $\overline{\mathbb{Z}}_{max}$, of which $\varepsilon = -\infty$ and $e = 0$.
The set $\overline{\mathbb{Z}}$ endowed with the *minimum* operator as sum and $+$ as product is a complete dioid, usually noted $\overline{\mathbb{Z}}_{min}$, of which $\varepsilon = +\infty$ and $e = 0$.

Definition 5. (Kleene star operator). Let * be the operator defined as follows: $a^* = \bigoplus_{i \in \mathbb{N}} a^i$, with $a^0 = e$.
This operation is consistent with the order $\succeq$ in the following sense: $\forall (a,b) \in \mathcal{D}^2$, if $a \succeq b$, then $a^* \succeq b^*$.

Theorem 6. (Baccelli et al. [1992]). Over a complete dioid, the implicit equation $x = ax \oplus b$ admits $x = a^*b$ as least solution.

Theorem 7. If $\mathcal{D}$ is a dioid, the set $\mathcal{D}^{n \times n}$ of $n \times n$ matrices and entries in $\mathcal{D}$ is also a dioid where the sum and the product are defined by: $\forall A, B \in \mathcal{D}^{n \times n}, (A \oplus B)_{ij} = A_{ij} \oplus B_{ij}$ and $(A \otimes B)_{ij} = \bigoplus_{k=1}^{n} A_{ik} \otimes B_{kj}$.

2.2 Dioids and interval arithmetics

We will briefly present how interval arithmetics can be applied to dioids. We recommend reading Litvinov and Sobolevskiĭ [2001] and Lhommeau et al. [2004] for further information.

Definition 8. (Interval). A (closed) interval in dioid $\mathcal{D}$ is a set of the form $\mathbf{x} = [\underline{x}, \overline{x}] = \{t \in \mathcal{D} | \ \overline{x} \succeq t \succeq \underline{x}\}$ where $(\underline{x}, \overline{x}) \in \mathcal{D}^2$ are called the lower and the upper bounds of the interval $\mathbf{x}$, respectively.

Definition 9. (Dioid of interval). A dioid of interval, denoted $I(\mathcal{D})$, can be defined out of a dioid $\mathcal{D}$ if it is endowed with two algebraic operations, $\overline{\oplus}$ and $\overline{\otimes}$ such that $\mathbf{x} \bar{\star} \mathbf{y} = [\underline{x} \star \underline{y}, \overline{x} \star \overline{y}], \forall \mathbf{x}, \mathbf{y} \in I(\mathcal{D})$. The intervals $\boldsymbol{\varepsilon} = [\varepsilon, \varepsilon]$ and $\mathbf{e} = [e, e]$ are neutral elements of $\overline{\oplus}$ and $\overline{\otimes}$, respectively.

Remark 10. Since $\overline{x} \star \overline{y} \succeq \underline{x} \star \underline{y}$ whenever $\overline{x} \succeq \underline{x}$ and $\overline{y} \succeq \underline{y}$, then $I(\mathcal{D})$ is closed w.r.t. the operations $\overline{\oplus}$ and $\overline{\otimes}$.

Definition 11. Let $\{\mathbf{x}_\alpha\}$ be an infinite subset of $I(\mathcal{D})$, the infinite sum of elements of this subset is:

$$\overline{\bigoplus_{\alpha}} \mathbf{x}_\alpha = [\bigoplus_{\alpha} \underline{x}_\alpha, \bigoplus_{\alpha} \overline{x}_\alpha]$$

Definition 12. (Order relation). Dioid $I(\mathcal{D})$ can be endowed with a natural (partial) order:
$\mathbf{a} \succeq_{I(\mathcal{D})} \mathbf{b} \Leftrightarrow \mathbf{a} = \mathbf{a} \overline{\oplus} \mathbf{b} \Leftrightarrow \overline{a} \succeq_{\mathcal{D}} \overline{b}$ and $\underline{a} \succeq_{\mathcal{D}} \underline{b}$

Theorem 13. (Litvinov and Sobolevskiĭ [2001]). If the dioid $\mathcal{D}$ is complete, then the dioid $I(\mathcal{D})$ is complete.

Remark 14. Note that if $\mathbf{x}$ and $\mathbf{y}$ are intervals in $I(\mathcal{D})$, then $\mathbf{x} \subset \mathbf{y} \Leftrightarrow \overline{y} \succeq \overline{x} \succeq \underline{x} \succeq \underline{y}$. In particular, $\mathbf{x} = \mathbf{y} \Leftrightarrow \underline{x} = \underline{y}$ and $\overline{x} = \overline{y}$.

Remark 15. $I(\mathcal{D})$ being closed with respect to the operations $\overline{\oplus}$ and $\overline{\otimes}$, the Kleene star operator admits a natural extension, thus $\mathbf{x}^* = \bigoplus_{i \in \mathbb{N}} \mathbf{x}^i = [\bigoplus_{i \in \mathbb{N}} \underline{x}^*, \bigoplus_{i \in \mathbb{N}} \overline{x}^*] = [\underline{x}^*, \overline{x}^*]$ with $\mathbf{x}^0 = \mathbf{e}$

2.3 Timed Event Graphs with time uncertainties

We now introduce what TEG's are, based on the Petri net formalism (see for example Murata [1989] for more information on this formalism).

Definition 16. (Timed Event Graph). An event graph is a Petri net of which places have exactly one upstream and one downstream transition. An event graph is said to be timed when to each place is associated a delay, defined in the set of natural numbers.

Remark 17. Let us note that *concurrency* phenomena cannot be represented in TEG's since the corresponding modelling is a place with multiple incoming and/or outgoing arcs (see for instance Figure 2). Nevertheless, TEG's are interesting for representing *synchronizations* and *delays* taking place in processes.

A delay assigned to a place expresses the *minimal* sojourn time of a token in this place. By applying the earliest firing rule, TEG's can be seen as linear discrete event dynamical systems by using some dioid algebras (Cohen et al. [1989], Baccelli et al. [1992]). For instance it is possible to obtain a linear state representation in $\overline{\mathbb{Z}}_{max}$, by associating with each transition x a dater $x(k) : \mathbb{Z} \to \overline{\mathbb{Z}}_{max}$, which is an increasing mapping representing the date of the k^{th} firing of transition x.

Many temporal uncertainties are embedded in the complex systems we study. A diod of intervals allows for linear modelling of uncertainties (Lhommeau et al. [2004]).

[1] The symbol $\otimes$ will be omitted when no confusion is possible with the traditional product

Linear relations in such an algebra can be mapped into a TEG with uncertainties and vice-versa. The intervals used as delays represent the minimal and the maximal compulsory sojourn time of a token, *before* it can actually be consumed in transition firings. The uncertain delays are settled *dynamically* when a token appears in a place having such a delay. Thus we model intervals of guaranteed behaviour, taking into account temporal uncertainties. There is a crucial difference between time Petri net (be the delays associated to transitions (Merlin [1974]) or to places (Khansa et al. [1996])) and the TEG with temporal uncertainties. In the latter, a token does not die when a supremum is reached, unlike in time Petri nets where the timing information represents hard constraints.

Example 18. Figure 1 may represent a flexible manufacturing cell behaving as follows: three pieces of material can be processed at a time and the processing takes between 2 and 5 units of time, depending on the tool to be used. If the workstation is not available, the pieces of material stay in the upstream stock S_{up}. When a processing is finished, the piece of material is put in the downstream stock S_{down} [2] and another piece of material can be processed at once, or after one unit of time if the tool is to be changed.

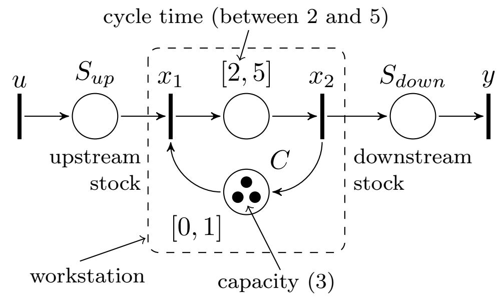

Figure 1. Workstation model

Transition x_1 is a synchronization. It will be fired only if both places S_{up} (a product has arrived in the workstation upstream stock) and C (the workstation is available) contain at least one token and when the uncertain delay over place C has been spent. By using the dater functions associated to the transitions of this TEG, we obtain:

$$\begin{cases} max(u(k), x_2(k-3)) \leq x_1(k) \leq max(u(k), 1 + x_2(k-3)) \\ 2 + x_1(k) \leq x_2(k) \leq 5 + x_1(k) \\ y(k) = x_2(k) \end{cases}$$

We can rewrite the first line of this system by $x_1(k) \in [u(k) \oplus x_2(k-3), u(k) \oplus 1 \otimes x_2(k-3)]$. This interval represent the extreme behaviours of transition $x_1(k)$, showing the best and worst cases. So in the dioid $I(\overline{\mathbb{Z}}_{max})$, this system turns into a system of linear state equations:

$$\begin{cases} \mathbf{x_1}(k) = \mathbf{u}(k) \oplus [0,1] \otimes \mathbf{x_2}(k-3) \\ \mathbf{x_2}(k) = [2,5] \otimes \mathbf{x_1}(k) \\ \mathbf{y}(k) = \mathbf{x_2}(k) \end{cases} \quad (1)$$

A dater may also be represented by its γ-transform, formally defined by $\gamma x(k) = x(k-1)$. γ may be regarded as the backward shift operator in the event domain. Thus daters may be turned into formal power series with coefficients in $\overline{\mathbb{Z}}_{max}$ and exponents in $\mathbb{Z}$, of the form $X(\gamma) = \bigoplus_{k \in \mathbb{Z}} x(k)\gamma^k$, in the dioid denoted $\overline{\mathbb{Z}}_{max}[\![\gamma]\!]$.

[2] both S_{up} and S_{down} have a capacity which is supposed infinite

In this dioid, linear state equation systems have the form of the following canonical system:

$$\begin{cases} X(\gamma) = AX(\gamma) \oplus BU(\gamma) & (2) \\ Y(\gamma) = CX(\gamma) & (3) \end{cases}$$

Where $X \in (\overline{\mathbb{Z}}_{max}[\![\gamma]\!])^n$ represents the internal transitions behaviour, $U \in (\overline{\mathbb{Z}}_{max}[\![\gamma]\!])^p$ represents the input transitions behaviour, and $Y \in (\overline{\mathbb{Z}}_{max}[\![\gamma]\!])^q$ represents the output transitions behaviour, and $A \in (\overline{\mathbb{Z}}_{max}[\![\gamma]\!])^{n \times n}$, $B \in (\overline{\mathbb{Z}}_{max}[\![\gamma]\!])^{n \times p}$ and $C \in (\overline{\mathbb{Z}}_{max}[\![\gamma]\!])^{q \times n}$ represent the link between the transitions.
The class of uncertain systems which are considered are TEG's where time delays are only known to belong to intervals. Therefore uncertainties can be described by intervals with known lower and upper bounds and the matrices of equations (2) and (3) are such that $A \in \mathbf{A} \in I(\overline{\mathbb{Z}}_{max}[\![\gamma]\!])^{n \times n}$, $B \in \mathbf{B} \in I(\overline{\mathbb{Z}}_{max}[\![\gamma]\!])^{n \times p}$ and $C \in \mathbf{C} \in I(\overline{\mathbb{Z}}_{max}[\![\gamma]\!])^{q \times n}$. Each entry of matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are intervals with bounds in dioid $\overline{\mathbb{Z}}_{max}[\![\gamma]\!]$ with only non-negative exponents and coefficients integer values. By theorem (6), equation (2) has the minimum solution $X = A^*BU$. Therefore, $Y = CA^*BU$ and the transfer function of the system is $H = CA^*B \in \mathbf{H} = \mathbf{CA}^*\mathbf{B} \in I(\overline{\mathbb{Z}}_{max}[\![\gamma]\!])^{q \times p}$, where $\mathbf{H}$ represents the interval in which the transfer function will lie for all the variations of the parameters.

Consequently, by introducing state vector $\mathbf{X}(\gamma) = (\mathbf{x_1}(\gamma)\ \mathbf{x_2}(\gamma))^t$, system (1) is equivalent in the dioid $I(\overline{\mathbb{Z}}_{max}[\![\gamma]\!])$ to the following system:

$$\begin{cases} \mathbf{X}(\gamma) = \begin{pmatrix} [\varepsilon,\varepsilon] & [\gamma^3, 1\gamma^3] \\ [2,5] & [\varepsilon,\varepsilon] \end{pmatrix} \mathbf{X}(\gamma) \oplus \begin{pmatrix} [e,e] \\ [\varepsilon,\varepsilon] \end{pmatrix} \mathbf{u}(\gamma) \\ \mathbf{y}(\gamma) = ([\varepsilon,\varepsilon]\ [e,e])\, \mathbf{X}(\gamma) \end{cases}$$

3. MUTUAL EXCLUSION

In this section we will show how to deal with mutual exclusion of shared resources sections, by turning this problem into an interval analysis problem.

3.1 Class of studied problems

In this article, we consider production systems with two workstations (namely W_1 and W_2) sharing one resource (called R), as depicted in Figure 2. Each workstation has its own upstream stock (namely S_1 and S_2) and entry point (u_1 and u_2 respectively) for new incoming material. At the end of their process, the workstations deliver the produced material in their own downstream stocks, from which they will leave the system through exit points y_1 and y_2.
The management rule we use for the shared resource is described in the next section and represented in Figure 3.

3.2 Dealing with shared resources

In this paper, shared resources between several workstations are handled according to the following rule:
If all workstations are idle, any incoming piece of material can be processed at once, on any workstation. Since all workstations share the same unique resource, only one workstation can be processing at a time. If more than one

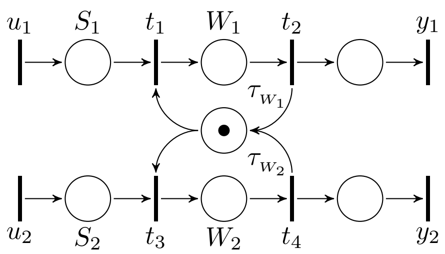

Figure 2. Shared resource section

workstation upstream stock is non empty when a workstation finishes its process, the workstations will seize the resource on a cyclic basis. If only one upstream stock is non empty, the incoming material will be processed on the corresponding machine as long as no other piece of material arrives in the upstream stock of the other workstation. This rule is illustrated in Figure 3 in the Petri net formalism, using inhibitor arcs (Reinhardt [1996]).

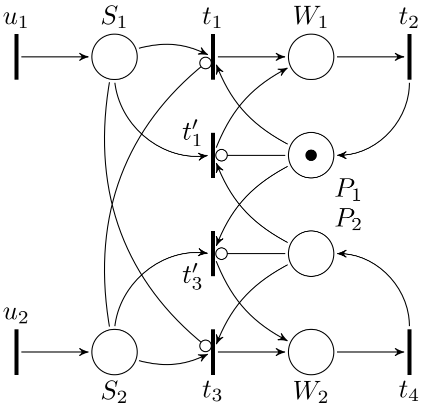

Figure 3. Principle diagram of our dispatching rule

Transitions t_1, t_1', t_3 and t_3' of Figure 3 illustrate the four possible cases, when a decision about the resource allocation has to be taken. If a token is in place P_1 (resp. P_2), then W_1 (resp. W_2) has been the last workstation to seize the resource and to process a piece of material.
If there is a token in P_1 and none in S_2, then W_1 can seize the resource, thus t_1 is fired. W_2 has the same behaviour if there is a token in P_2 and none in S_1 (t_3 is fired in that case). If there is at least one token in S_1 and S_2 at the same time, then the resource is alternatively seized by the two workstations (either t_1' or t_3' is fired depending on the last workstation to seize the shared resource).

3.3 Properties of the workstations

We will now describe the behaviour of the workstations. Based upon the Petri net depicted in Figure 2, we can make several statements about the firing of the transitions.

Using the assignment policy presented in the last subsection, it obviously appears that when there is no conflict for seizing the resource, the behaviour between the entry u_i and the corresponding exit y_i is $\overline{\mathbb{Z}}_{max}$ linear, for all i.
If the shared resource is strongly requested (there is a lot of tokens in S_1 or in S_2), then it is alternatively assigned to the two production lines. Therefore, the head token in S_1 or in S_2 (the first one which arrived) waits at most $\tau_{P_1} + \tau_{P_2}$ units of time.
These two cases describe the dynamic of the system in the best case (no conflict) and the worst case (a lot of conflicts).

Proposition 19. Let $u_1(k)$, $u_2(k)$, $t_1(k)$, $t_2(k)$, $t_3(k)$ and $t_4(k)$ be the k^{th} firing dates of transitions u_1, u_2, t_1, t_2, t_3 and t_4 of Figure 3.
$\forall k \in \mathbb{N}$ and *according to our production management rule*, we have:

$$\begin{cases} max(u_1(k), t_2(k-1)) \leq t_1(k) & (4)\\ t_1(k) \leq max(u_1(k) + \tau_{W_2}, t_2(k-1) + \tau_{W_2}) & (5)\\ max(u_2(k), t_4(k-1)) \leq t_3(k) & (6)\\ t_3(k) \leq max(u_2(k) + \tau_{W_1}, t_4(k-1) + \tau_{W_1}) & (7)\end{cases}$$

Proof: By using the dater function associated to the transitions of Figure 2, and considering the earliest firing rule, we obtain the following system:

$$\begin{cases} t_1(k) \geq u_1(k) \oplus \bigoplus_{i+j=k+n} \{t_2(i-1) \oplus t_4(j-1)\} & (8)\\ n = \bigoplus_{x\in\mathbb{Z}} \{t_3(x) < +\infty\} & (9)\end{cases}$$

Say $k = 1$, then $\forall n \in \mathbb{N}$, $t_4(n-1) > t_2(k-1)$, thus $t_1(1) \geq max(u_1(1), t_4(n-1)) \geq max(u_1(1), t_2(k-1))$.
Hence (4) holds for $k = 1$.
If $k > 1$, then either $t_2(k-1) \geq t_4(n-1)$ and so $t_1(k) \geq max(u_1(k), t_2(k-1))$ (thanks to equation (8)), or $t_2(k-1) < t_4(n-1)$ and so $t_1(k) \geq max(u_1(k), t_4(n-1)) \geq max(u_1(k), t_2(k-1))$.
Hence (4) holds $\forall k$.
Now, comparing $u_1(k)$ and $t_2(k-1)$, we either have $u_1(k) \geq t_2(k-1)$ or $u_1(k) < t_2(k-1)$. In the first case, if $t_3(n-1) < u_1(k)$ (which means that W_2 is already processing material while some other material arrive in the upstream stock of W_1), then transition t_1 will be fired *as soon as* W_2 has finished processing, i.e. after τ_{W_2} units of time, on latest, after some material has arrived in S_1 (after u_1 has been fired). Hence $t_1(k) \leq u_1(k) + \tau_{W_2}$ in that case. If $t_3(n-1) \geq u_1(k)$, then no workstation has undertaken any process before u_1 has been fired, because all the incoming material has already been processed. So t_1 can be fired at once, i.e. $t_1(k) = u_1(k)$
If $u_1(k) < t_2(k-1)$, then either no material is "waiting" upstream of W_2 when W_1 has finished processing and t_1 is fired at once (thus $t_1(k) = t_2(k-1)$), or W_1 has to wait for the turn of W_2 to seize the resource first, i.e. $t_1(k) = t_4(n-1) = t_3(n-1) + \tau_{W_2} = t_2(k-1) + \tau_{W_2}$, which leads to the results of (5).
We would show that (6) and (7) hold true with the same reasonings on the symmetric transitions. □

Therefore, by using $\overline{\mathbb{Z}}_{max}$ notations applied to the production system depicted in Figure 2 and the production management rule defined in the previous subsection, we obtain:

$$\begin{cases} t_1(k) = u_1(k) \otimes \tau_a \oplus t_2(k-1) \otimes \tau_b\\ t_2(k) = \tau_{W_1} \otimes t_1(k)\\ t_3(k) = u_2(k) \otimes \tau_c \oplus t_4(k-1) \otimes \tau_d\\ t_4(k) = \tau_{W_2} \otimes t_3(k)\end{cases} \quad (10)$$

where $\tau_a, \tau_b \in [0, \tau_{W_2}]$ and $\tau_c, \tau_d \in [0, \tau_{W_1}]$.

3.4 From shared resources to parallel systems

Based on the intervals given in system (10), we can turn the model of Figure 2 in a slightly different model, as depicted in Figure 4. The original Petri net containing a shared resource is actually split into two TEG's. Manipulating TEG's will allow us for using dioid algebras.

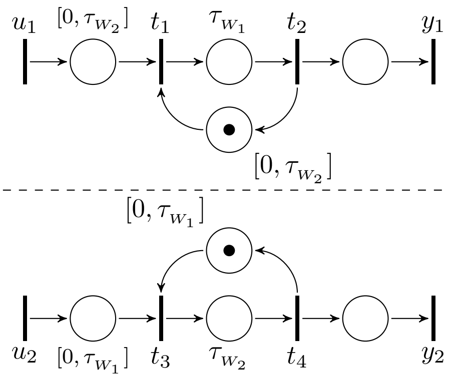

Figure 4. From shared resource to parallel Petri nets

Then, thanks to the dioid of intervals, we can actually synthesize the dynamic behaviour of the production system modelled by those TEG's.
Of course, the goal here is not to provide the actual transfer function of the system, but to enclose it in an interval, so that we can do formal calculations on the latter, which would be *impossible* with an accurate representation.

Using the modelling form of equations (2) and (3), we obtain the following interval matrices for workstation 1, which is depicted in the upper part of Figure 4:

$$\mathbf{A} = \begin{pmatrix} [\varepsilon, \varepsilon] & [\gamma, \tau_{W_2}\gamma] \\ [\tau_{W_1}, \tau_{W_1}] & [\varepsilon, \varepsilon] \end{pmatrix}$$
$$\mathbf{B} = \begin{pmatrix} [0, \tau_{W_2}] \\ [\varepsilon, \varepsilon] \end{pmatrix}$$
$$\mathbf{C} = \left([\varepsilon, \varepsilon] \ [e, e]\right)$$

The interval matrix $\mathbf{H_1}$ given below includes the actual transfer function H_1 of the sub-system representing workstation 1. The former characterizes all of its possible behaviours.

$$\mathbf{H_1} = \mathbf{C}\mathbf{A}^*\mathbf{B} = \left([\tau_{W_1}(\tau_{W_1}\gamma)^*, \tau_{W_1}\tau_{W_2}(\tau_{W_1}\tau_{W_2}\gamma)^*]\right)$$

Considering the input/output relation $y(\gamma) = \mathbf{H}u(\gamma)$, where $u(\gamma) = (u_1(\gamma)\ u_2(\gamma))^t$ and $y(\gamma) = (y_1(\gamma)\ y_2(\gamma))^t$, we obtain the following transfer matrix $\mathbf{H}$ for the whole production system:

$$\mathbf{H} = \begin{pmatrix} H_1 & \varepsilon \\ \varepsilon & H_2 \end{pmatrix}$$

where $\mathbf{H_2} = \left([\tau_{W_2}(\tau_{W_2}\gamma)^*, \tau_{W_1}\tau_{W_2}(\tau_{W_1}\tau_{W_2}\gamma)^*]\right)$.

4. APPLICATION CASE

We have applied our method on a complex automated transfer line located in the LISA laboratory, in Angers, France. This transfer line includes an "eight shape" (see Figure 5), which implies the use of a management rule in the common sections. The first junction (hatched pattern in the figure) can be considered as a shared resource. Indeed, only one pallet can be in this section at a time, to ensure that pallets do not overlap. The second junction does not need any management rule in our case since all pallets have already been distanced enough in the common section (in gray in the figure).
All pallets are loaded at a loading point, and flow in the system until the unloading point, where they are unloaded. Their path follows the direction of the numbered arrows in the increasing order. The matter here is to anticipate the flow of pallets in the first junction, for instance by using the production management rule defined in section 3.2.

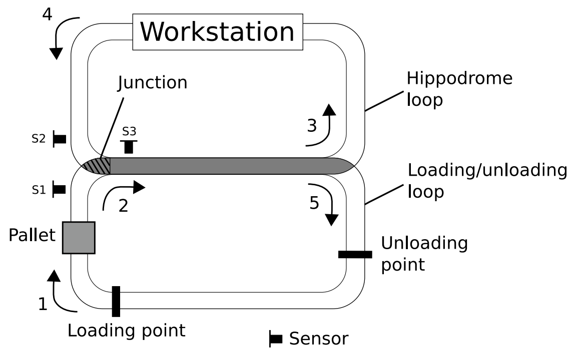

Figure 5. Automated transfer line

We can consider the junction between the three sensors depicted in the figure as being a shared resource, and the paths from $S1$ to $S3$ and from $S2$ to $S3$ can be seen as workstations. In that way, we can represent the whole system by the diagram of Figure 6.

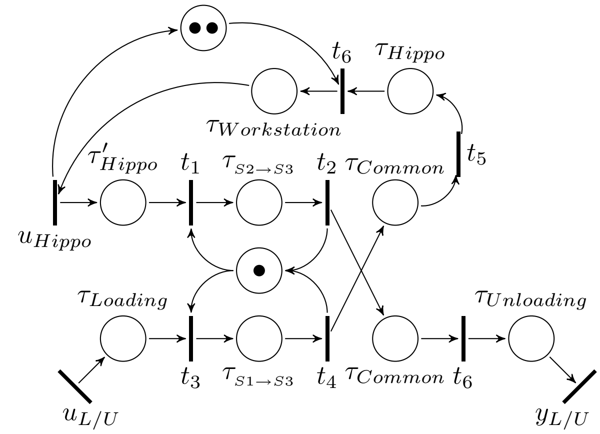

Figure 6. Petri net model of the transfer line

The decision of going either to the hippodrome or to the loading/unloading loops at the end of the common section is taken right from the first junction of the production system. Hence the two theoretical paths going out of the latter, having the same duration, τ_{Common}, equal to 10 units of time here.
The transportation times between $S1$ and $S3$ and between $S2$ and $S3$ (denoted respectively $\tau_{S1\to S3}$ and $\tau_{S2\to S3}$ in Figure 6) are equal to 6 and 7 units of time respectively [3].
The transfer times from the loading point up to the first junction and from the second junction up to the unloading

[3] they happen to be different because the upstream sensors of the first junction ($S1$ and $S2$) are not exactly located at the same distance from the downstream sensor ($S3$)

point, called $\tau_{Loading}$ and $\tau_{Unloading}$, both take 20 units of time. The transfer time τ_{Hippo} on the hippodrome from the common section to the workstation and the transfer time τ'_{Hippo} from the workstation to the common section take respectively 20 and 25. The workstation can process two pieces of material at a time and this process takes 30 units of time ($\tau_{Workstation}$) for each piece. By denoting $\boldsymbol{\tau_x}$ the interval of which both bounds are equal to τ_x and $\mathbf{H_{L/U}}$ and $\mathbf{H_{Hippo}}$ the transfer functions of the first junction parts, belonging respectively to the loading/unloading loop and to the hippodrome loop, we find the following transfer function $\mathbf{H}$ for the whole system in the dioid $I(\overline{\mathbb{Z}}_{max}[\![\gamma]\!])$:

$$\begin{aligned}\mathbf{H} = \ & \boldsymbol{\tau}_{\mathbf{Loading}} \overline{\otimes} \mathbf{H_{L/U}} \overline{\otimes} \boldsymbol{\tau}_{\mathbf{Common}} \overline{\otimes} \boldsymbol{\tau}_{\mathbf{Hippo}} \overline{\otimes} \\ & \boldsymbol{\tau}_{\mathbf{Workstation}} \overline{\otimes} \boldsymbol{\tau}_{\mathbf{Hippo}}{}' \overline{\otimes} \mathbf{H_{Hippo}} \overline{\otimes} \\ & \boldsymbol{\tau}_{\mathbf{Common}} \overline{\otimes} \boldsymbol{\tau}_{\mathbf{Unloading}}\end{aligned}$$

where

$$\begin{aligned}\mathbf{H_{L/U}} &= ([\tau_{S1\to S3}(\tau_{S1\to S3}\gamma)^*, \\ &\qquad \tau_{S1\to S3}\tau_{S2\to S3}(\tau_{S1\to S3}\tau_{S2\to S3}\gamma)^*]) \\ &= ([6(6\gamma)^*, 13(13\gamma)^*])\end{aligned}$$

and

$$\mathbf{H_{Hippo}} = ([7(7\gamma)^*, 13(13\gamma)^*]) \,.$$

Finally, we obtain

$$\begin{aligned}\mathbf{H} &= [20,20]\overline{\otimes}[6(6\gamma)^*, 13(13\gamma)^*]\overline{\otimes}[10,10]\overline{\otimes} \\ &\quad [20,20]\overline{\otimes}[30(30\gamma^2)^*, 30(30\gamma^2)^*]\overline{\otimes}[25,25]\overline{\otimes} \\ &\quad [7(7\gamma)^*, 13(13\gamma)^*]\overline{\otimes}[10,10]\overline{\otimes}[20,20] \\ &= [115,115]\overline{\otimes}[13(7\gamma)^*, 26(13\gamma)^*]\overline{\otimes} \\ &\quad [30(30\gamma^2)^*, 30(30\gamma^2)^*] \\ &= ([(158\oplus 165\gamma)(30\gamma^2)^*, (171\oplus 184\gamma)(30\gamma^2)^*])\end{aligned}$$

This expression expresses embedded several aspects of the transfer line. For instance, the production rate of the studied system is 2 products every 30 units of time. Thus we figure the behaviour uncertainty of the shared resource section ($[13(7\gamma)^*, 26(13\gamma)^*]$) is somehow hidden by the behaviour of the workstation ($30(30\gamma^2)^*$). Moreover, the first product leaving this manufacturing unit will flow in the system during between 158 and 171 units of time.

5. CONCLUSION

Based on the DES's paradigm, this research has attempted to describe a shared resource problem in the $I(\overline{\mathbb{Z}}_{max}[\![\gamma]\!])$ dioid, the dioid of interval with bounds in dioid of formal power series with coefficients in $\overline{\mathbb{Z}}_{max}$ and exponents in $\mathbb{Z}$. The behaviours of the studied systems are confined in intervals of $I(\overline{\mathbb{Z}}_{max}[\![\gamma]\!])$ so as to obtain a formal representation of them.
An application case consisting of a eight shaped loop has been used to put our method to the test. The results seem promising for future modelling and control developments on similar cases.

Synthesizing a (feedback) controller thanks to the interval formal representation has already been achieved, but has not been included in this article for page limitation reasons. This controller help avoiding deadlocks in such a system and lead to just-in-time management [4], by limiting the number of transporters in it, in case it would become too overloaded. The main advantage of this approach is the use of the same formalism for both system and controller modelling. Moreover, the latter being expressed in a formal way, testing it through emulation is very reliable (Boutin et al. [2007]).
The next step is to extend this work not only to a shared resource between workstations, but between whole $\overline{\mathbb{Z}}_{max}$ linear systems.

[4] i.e. achieving some performance while minimizing internal stocks

REFERENCES

Michel Al Saba, Jean-Louis Boimond, and Sébastien Lahaye. On just in time control of flexible manufacturing systems via dioid algebras. *INCOM'06*, Saint-Étienne, France, 2006.

Saïd Amari, Isabel Demongodin, and Jean-Jacques Loiseau. *Supply Chain Optimization*, chapter 6 - Sizing, Cycle Time and Plant Control Using Dioid Algebra, pages 71 – 86. Springer Verlag, 2004.

François Baccelli, Guy Cohen, Geert Jan Olsder, and Jean-Pierre Quadrat. *Synchronization and Linearity, An Algebra for Discrete Event Systems.* John Wiley and Sons, New York, 1992. available at `cermics.enpc.fr/~cohen-g/documents/BCOQ-book.pdf`.

Olivier Boutin, Bertrand Cottenceau, and Anne L'Anton. Online Control of a (max,+) Linear Emulated Production System. *IESM'07*, Beijing, China, 2007. Tsinghua University Press.

Guy Cohen, Pierre Moller, Jean-Pierre Quadrat, and Michel Viot. Algebraic Tools for the Performance Evaluation of Discrete Event Systems. *IEEE Proceedings: special issue on Discrete Event Systems*, 77(1):39 – 58, 1989.

Stéphane Gaubert. Performance Evaluation of (max,+) automata. *IEEE Transactions on Automatic Control*, 40(12), December 1995.

Bernd Heidergott, Geert Jan Olsder, and Jacob 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.

Wael Khansa, Jean-Paul Denat, and Simon Collart-Dutilleul. P-time petri nets for manufacturing systems. *WODES'96*, Edinburgh, Scotland, 1996.

Mehdi Lhommeau, Laurent Hardouin, Bertrand Cottenceau, and Luc Jaulin. Interval Analysis and Dioid: Application to Robust Controller Design for Timed Event Graphs. *Automatica*, 40(11):1923 – 1930, November 2004.

Grigory L. Litvinov and Andreĭ N. Sobolevskiĭ. Idempotent Interval Analysis and Optimization Problems. *Reliable Computing*, 7(5):353 – 377, 2001.

Philip M. Merlin. *A Study of the Recoverability of Computing Systems.* PhD thesis, University of California, Irvine, 1974.

Tadao Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541 – 580, April 1989.

Klaus Reinhardt. Reachability in Petri Nets with Inhibitor arcs. Technical Report WSI-96-30, Wilhelm-Schickard Institut für Informatik, Universität Tübingen, 1996.

Benoît Trouillet and Ahmer Benasser. Cyclic Scheduling Problems With Assemblies: An Approach Based to the Search of an Initial Marking in a Marked Graph. *SMC'02*, Hammamet, Tunisia, 2002. IEEE Computer Society Press.