



**HAL**  
open science

## Discrete Event System Control in Max-Plus Algebra : Application to Manufacturing Systems

Gabriel Freitas Oliveira, Renato Markele Ferreira Cândido, Vinicius Mariano Gonçalves, Carlos Andrey Maia, Bertrand Cottenceau, Laurent Hardouin

► **To cite this version:**

Gabriel Freitas Oliveira, Renato Markele Ferreira Cândido, Vinicius Mariano Gonçalves, Carlos Andrey Maia, Bertrand Cottenceau, et al.. Discrete Event System Control in Max-Plus Algebra : Application to Manufacturing Systems. 15th IFAC Workshop on Discrete Event Systems, Nov 2020, Rio de Janeiro, Brazil. hal-03223212

**HAL Id: hal-03223212**

**<https://univ-angers.hal.science/hal-03223212>**

Submitted on 24 May 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

# Discrete Event System Control in Max-Plus Algebra : Application to Manufacturing Systems\*

Gabriel Freitas Oliveira \* Renato Markele Ferreira Candido \*  
Vinicius Mariano Gonçalves \*\* Carlos Andrey Maia \*\*  
Bertrand Cottenceau \* Laurent Hardouin \*

\* *Laboratoire Angevin de Recherche en Ingénierie des Systèmes,  
Université d'Angers, France (e-mails :  
laurent.hardouin@univ-angers.fr)*

\*\* *Departamento de Engenharia Elétrica, Universidade Federal de  
Minas Gerais, Belo Horizonte, Brazil*

---

**Abstract:** This paper deals with control of industrial systems which can be depicted by timed event graphs. A methodology and software tools are presented to allow engineers to implement automatically controllers in PLC. Three steps are needed and recalled: the modelling of the system in max plus algebra, the design of the controller and the implementation in a Supervisory Control and Data Acquisition (SCADA) system. An example on a real system is given to illustrate these steps.

*Keywords:* Petri nets, timed event graphs, max-plus algebra.

---

## 1. INTRODUCTION

The industrial manufacturing systems are frequently event-driven systems and their closed-loop control is worth of interest in order to react to disturbances modifying their behaviours. Among these discrete event systems there exists a subclass of system involving synchronization and delay phenomena for which an efficient control theory has been developed during the last decade (Schutter et al., 2019; Maia et al., 2005; Lhommeau et al., 2005; Shang et al., 2016; Heidergott et al., 2006; Cohen et al., 1998). Synchronization phenomena appear when meeting between events is needed, e.g. an event starts when all the preceding events are finished. Delay phenomena appear when transportation operation or manufacturing activities need a duration to be achieved, e.g. the event representing a finishing time of a task is equal to its starting time plus the duration of the operation. Although these phenomena lead to a non-linear model in conventional algebra they admit a linear model formulated in some idempotent semirings, the most popular being the max-plus algebra, where the max operation is the sum of the algebraic structure and the classical addition has to be considered as the product. It must be noticed that these systems and their linear models correspond exactly to the Timed Event Graphs (TEGs) which are a subclass of Petri nets where each place has exactly one upstream and one downstream transition and all arcs have weight equal to 1. The time is associated to places and represents the duration a token has to stay in a place before to contribute to the firing of a downstream transition. This graphical model is popular since suitable for engineer depicting the behaviour of a manufacturing system. This paper proposes a methodology and a specific

software to help the engineers to synthesize and implement a closed-loop control of these systems by tacking advantage of the efficient existing control strategies. By starting from a TEG description of the system and a desired behaviour also given as a TEG named reference model, a method is proposed to obtain the algebraic model of the system and of the reference model, then a software is proposed to give automatically the code to implement in a Supervisory Control and Data Acquisition (SCADA) system. The engineer will have to choose the control strategy which will be implemented among different strategies, namely the disturbance decoupling problem, the model matching problem, the observer-based controller. They have for common point to yield an optimal control law according to the just-in-time criterion which aims to delay as much as possible the occurrence of event input while achieving the specific goal of the control strategy. This criterion is quite usual in industry since it leads to produce only the necessary quantity of raw materials and then to reduce as much as possible the useless stock.

In this paper we will recall the observer-based controller strategy in order to illustrate the methodology used in the software. It is organized as follows: First, the algebraic tools necessary to synthesize the control law are recalled, then in section 3 the method to obtain the algebraic model of the system is presented, this section presents an original description which allows the software to obtain automatically an explicit model. The observer-based control strategy which aims to match a reference model is presented in section 4. The software architecture and the methodology adopted are illustrated in section 5, which presents an implementation on a flexible automated system available at University of Angers.

---

\* CAPES-COFECUB

## 2. MATHEMATICAL BACKGROUND

An idempotent semiring  $\mathcal{S}$  is an algebraic structure with two internal operations denoted by  $\oplus$  and  $\otimes$ . The operation  $\oplus$  is associative, commutative and idempotent, that is,  $a \oplus a = a$ . The operation  $\otimes$  is associative (but not necessarily commutative) and distributive on the left and on the right with respect to  $\oplus$ . The neutral elements of  $\oplus$  and  $\otimes$  are represented by  $\varepsilon$  and  $e$  respectively, and  $\varepsilon$  is an absorbing element for the law  $\otimes$  ( $\forall a \in \mathcal{S}, \varepsilon \otimes a = a \otimes \varepsilon = \varepsilon$ ). As in classical algebra, the operator  $\otimes$  will be often omitted in the equations, moreover,  $a^i = a \otimes a^{i-1}$  and  $a^0 = e$ . In this algebraic structure, a partial order relation is defined by  $a \succeq b \Leftrightarrow a$ , therefore an idempotent semiring  $\mathcal{S}$  is a partially ordered set (see (Baccelli et al., 1992; Heidergott et al., 2006) for an exhaustive introduction). An idempotent semiring  $\mathcal{S}$  is said to be complete if it is closed for infinite  $\oplus$ -sums and if  $\otimes$  distributes over infinite  $\oplus$ -sums. In particular  $\top = \bigoplus_{x \in \mathcal{S}} x$  is the greatest element of  $\mathcal{S}$  ( $\top$  is called the top element of  $\mathcal{S}$ ).

*Theorem 1.* [see Baccelli et al. (1992), th. 4.75] The implicit inequality  $x \succeq ax \oplus b$  as well as the equation  $x = ax \oplus b$  defined over  $\mathcal{S}$ , admit  $x = a^*b$  as the least solution, where  $a^* = \bigoplus_{i \in \mathbb{N}} a^i$  (Kleene star operator).

*Definition 1.* (Residual and residuated mapping). An order preserving mapping  $f : \mathcal{D} \rightarrow \mathcal{E}$ , where  $\mathcal{D}$  and  $\mathcal{E}$  are partially ordered sets, is a *residuated mapping* if for all  $y \in \mathcal{E}$  there exists a greatest solution for the inequality  $f(x) \preceq y$  (hereafter denoted  $f^\sharp(y)$ ). Obviously, if equality  $f(x) = y$  is solvable,  $f^\sharp(y)$  yields the greatest solution. The mapping  $f^\sharp$  is called the *residual* of  $f$  and  $f^\sharp(y)$  is the optimal solution of the inequality.

*Example 1.* Mappings  $\Lambda_a : x \mapsto a \otimes x$  and  $\Psi_a : x \mapsto x \otimes a$  defined over an idempotent semiring  $\mathcal{S}$  are both residuated (Baccelli et al. (1992), p. 181). Their residuals are order preserving mappings denoted respectively by  $\Lambda_a^\sharp(x) = a \backslash x$  and  $\Psi_a^\sharp(x) = x / a$ . This means that  $a \backslash b$  (resp.  $b / a$ ) is the greatest solution of the inequality  $a \otimes x \preceq b$  (resp.  $x \otimes a \preceq b$ ).

The set of  $n \times n$  matrices with entries in  $\mathcal{S}$  is an idempotent semiring. The sum, the product and the residuation of matrices are defined after the sum, the product and the residuation of scalars in  $\mathcal{S}$ , *i.e.*,

$$(A \otimes B)_{ik} = \bigoplus_{j=1..n} (a_{ij} \otimes b_{jk}) \quad (1)$$

$$(A \oplus B)_{ij} = a_{ij} \oplus b_{ij}, \quad (2)$$

$$(A \backslash B)_{ij} = \bigwedge_{k=1..n} (a_{ki} \backslash b_{kj}), \quad (B / A)_{ij} = \bigwedge_{k=1..n} (b_{ik} / a_{jk}). \quad (3)$$

The identity matrix of  $\mathcal{S}^{n \times n}$  is the matrix with entries equal to  $e$  on the diagonal and to  $\varepsilon$  elsewhere. This identity matrix will also be denoted  $e$ , and the matrix with all its entries equal to  $\varepsilon$  will also be denoted  $\varepsilon$ .

*Example 2.* (max,plus) algebra:  $\overline{\mathbb{Z}}_{\max} = (\mathbb{Z} \cup \{-\infty, +\infty\}, \max, +)$  is a complete idempotent semiring such that  $a \oplus b = \max(a, b), a \otimes b = a + b, a \wedge b = \min(a, b)$  with

$\varepsilon = -\infty, e = 0$ , and  $\top = +\infty$ . The order  $\preceq$  is total and corresponds to the natural order  $\leq$ . By extension  $\overline{\mathbb{Z}}_{\max}^{n \times n}$  is a semiring of matrices with entries in  $\overline{\mathbb{Z}}_{\max}$ . Matrix  $\varepsilon \in \overline{\mathbb{Z}}_{\max}^{n \times m}$  will be such that all its entries are equal to  $\varepsilon \in \overline{\mathbb{Z}}_{\max}$ , matrix  $e \in \overline{\mathbb{Z}}_{\max}^{n \times n}$  will be such that all the entries are equal to  $e \in \overline{\mathbb{Z}}_{\max}$  except the diagonal entries which are equal to  $e \in \overline{\mathbb{Z}}_{\max}$ .

*Example 3.* (min,plus) algebra:  $\overline{\mathbb{Z}}_{\min} = (\mathbb{Z} \cup \{-\infty, +\infty\}, \min, +)$  is a complete idempotent semiring such that  $a \oplus b = \min(a, b), a \otimes b = a + b, a \wedge b = \max(a, b)$  with  $\varepsilon = +\infty, e = 0$ , and  $\top = -\infty$ . The order  $\preceq$  is total and corresponds to the inverse of the natural order (*i.e.*,  $2 \preceq 1$ ). Semiring of matrices  $\overline{\mathbb{Z}}_{\min}^{n \times n}$  is a semiring of matrices with entries in  $\overline{\mathbb{Z}}_{\min}$ .

*Example 4.* (Matrix operations in  $\overline{\mathbb{Z}}_{\max}$ ). Given three matrices with entries in  $\overline{\mathbb{Z}}_{\max}$ ,

$$A = \begin{bmatrix} 1 & 4 \\ 5 & 3 \\ \varepsilon & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 3 \\ 2 & 4 \\ 7 & 1 \end{bmatrix}, \quad \text{and} \quad C = \begin{bmatrix} \varepsilon & 4 \\ 1 & 3 \end{bmatrix}$$

we get

$$A \oplus B = \begin{bmatrix} 1 & 4 \\ 5 & 3 \\ \varepsilon & 2 \end{bmatrix} \oplus \begin{bmatrix} 3 & 3 \\ 2 & 4 \\ 7 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 5 & 4 \\ 7 & 2 \end{bmatrix}$$

$$A \otimes C = \begin{bmatrix} 1 & 4 \\ 5 & 3 \\ \varepsilon & 2 \end{bmatrix} \otimes \begin{bmatrix} \varepsilon & 4 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 5 & 7 \\ 4 & 9 \\ 3 & 5 \end{bmatrix}$$

Considering the relation  $A \otimes X \preceq B$  with

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & \varepsilon \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 6 \\ 7 \\ 8 \end{bmatrix}$$

being matrices with entries in  $\overline{\mathbb{Z}}_{\max}$ . As the max-plus multiplication corresponds to the classical addition, its residual corresponds to conventional subtraction, *i.e.*,  $1 \otimes x \preceq 4$  admits the solution set  $\mathcal{X} = \{x | x \preceq 1 \backslash 4\}$  with  $1 \backslash 4 = 4 - 1 = 3$  being the greatest solution of this set. Applying the rules of residuation in max-plus algebra to the relation  $A \otimes X \preceq B$  results in:

$$A \backslash B = \begin{bmatrix} 1 \backslash 6 \wedge 3 \backslash 7 \wedge 5 \backslash 8 \\ 2 \backslash 6 \wedge 4 \backslash 7 \wedge \varepsilon \backslash 8 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

Matrix  $A \backslash B = [3 \ 3]^T$  is the greatest solution for  $X$  which ensures  $A \otimes X \preceq B$ . Indeed,

$$A \otimes (A \backslash B) = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} 3 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ 8 \end{bmatrix} \preceq \begin{bmatrix} 6 \\ 7 \\ 8 \end{bmatrix} = B.$$

*Remark 1.* Note that residuation achieves equality if a solution exists.

### 3. SYSTEM MODELING

#### 3.1 Dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$

Dioid  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$  (see (Baccelli et al., 1992; Hardouin et al., 2018)) is formally the quotient dioid of  $\mathbb{B}[\gamma, \delta]$  (the set of formal power series in two commutative variables  $\gamma$  and  $\delta$ , with Boolean coefficients and with exponents in  $\mathbb{Z}$ ), by the equivalence relation  $x\mathcal{R}y \leftrightarrow \gamma^*(\delta^{-1})^*x = \gamma^*(\delta^{-1})^*y$ . Dioid  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$  is complete.

As  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$  is a quotient dioid, an element of  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$  may admit several representatives in  $\mathbb{B}[\gamma, \delta]$ . The representative which is minimal with respect to the number of terms is called the minimum representative.

A simple geometrical interpretation of the previous equivalence relation is available in the  $(\gamma, \delta)$ -plane. Consider a monomial  $\gamma^k\delta^t \in \mathbb{B}[\gamma, \delta]$ , its south-east cone is defined as  $\{(k', t') | k' \geq k \text{ and } t' \leq t\}$ . The south-east cone of a series in  $\mathbb{B}[\gamma, \delta]$  is defined as the union of the south-east cones associated with the monomials composing the considered series. For two elements  $s_1$  and  $s_2$  in  $\mathbb{B}[\gamma, \delta]$ ,  $s_1\mathcal{R}s_2$  (i.e.,  $s_1$  and  $s_2$  are equal in  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ ) is equivalent to the equality of their south-east cones. Direct consequences of previous geometrical interpretation are:

- simplification rules in  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ 

$$\gamma^k \oplus \gamma^t = \gamma^{\min(k,t)} \text{ and } \delta^k \oplus \delta^t = \delta^{\max(k,t)} \quad (4)$$
- a simple formulation of the order relation for monomials
$$\gamma^n\delta^t \preceq \gamma^{n'}\delta^{t'} \Leftrightarrow n \geq n' \text{ and } t \leq t'$$

A simple interpretation of the variable  $\gamma$  and  $\delta$  for daters is available:

- multiplying a series  $s$  by  $\gamma$  is equivalent to shifting the argument of the associated dater function by -1.
- multiplying a series  $s$  by  $\delta$  is equivalent to shifting the values of the associated dater function by 1

*Example 5.* Consider the series  $s = \gamma\delta^2 \oplus \gamma^3\delta^3 \oplus \gamma^4\delta^1$  represented by dots in Fig. 1. The minimum representative of  $s$  in  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$  is  $\gamma\delta^2 \oplus \gamma^3\delta^3$ . This result could be obtained using the simplification rules (4). Besides,

$$s = \bigoplus_{k \leq 0} \gamma^k \delta^{-\infty} \oplus \bigoplus_{k=1,2} \gamma^k \delta^2 \oplus \bigoplus_{k \geq 3} \gamma^k \delta^3$$

Therefore, the dater  $d_s$  associated with  $s$  is given by

$$d_s(k) = \begin{cases} -\infty & \text{if } k \leq 0 \\ 2 & \text{if } k = 1, 2 \\ 3 & \text{if } k \geq 3 \end{cases}$$

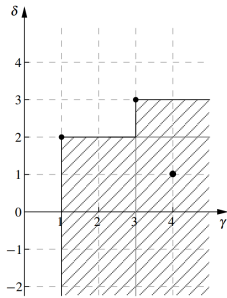


Fig. 1.  $s$  and its south-east cone (hatched)

#### 3.2 Linear state-space representation of TEG in $\mathcal{M}_{in}^{ax}[\gamma, \delta]$

From now on, we only consider TEG with at most one place from a transition to another transition. This assumption is not restrictive, as it is always possible to transform any TEG in an equivalent TEG with at most one place from a transition to another transition. The dynamics of a TEG may be captured by associating each transition with a series  $s \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ , where  $d_s(k)$  is defined as the time of firing  $k$  of the transition. Therefore, for TEG,  $\gamma$  is a shift operator in the event domain, where an event is interpreted as the firing of the transition, and  $\delta$  is a shift operator in the time domain.

The transitions of a TEG are divided into three categories:

- state transitions  $(x_1, \dots, x_n)$ : transitions with at least one input place and one output place.
- input transitions  $(u_1, \dots, u_p)$ : transitions with at least one output place, but no input places.
- output transitions  $(y_1, \dots, y_m)$ : transitions with at least one input place, but no output places.

Under the earliest functioning rule (i.e., state and output transitions fire as soon as they are enabled), with respect to a place with initially  $m$  tokens and holding time  $t$ , the influence of its upstream transition on its downstream transition is a positive shift in the time domain of  $t$  time units and a negative shift in the event domain of  $m$  events. The complete shift operator is coded by the monomial  $\gamma^m\delta^t$  in  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ . Therefore, consider the place upstream from transition  $t_i$  and downstream from transition  $t_j$ , the influence of transition  $t_j$  on transition  $t_i$  is coded by the monomial  $f_{ij}$  in  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$  defined by  $f_{ij} = \gamma^{m_{ij}}\delta^{\tau_{ij}}$  where  $m_{ij}$  is the initial number of tokens in the place and  $\tau_{ij}$  is the holding time of the place.

Consequently, a TEG admits a linear state-space representation in  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ .

$$\begin{cases} x = Ax \oplus Bu \oplus Rw \\ y = Cx \end{cases}$$

where  $x \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^n$  is the state,  $u \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^p$  the input,  $y \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^m$  the output and  $w \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^n$  the additive perturbation of the state. The perturbation  $w$  models, for example, unexpected failure, delays or uncertain parameters such as task duration and matrix  $R$  how these perturbations affect the inner states, in the sequel  $R$  is assumed to be the Identity matrix.  $A \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times n}$ ,  $B \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times p}$ ,  $C \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{m \times n}$  and  $R \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times n}$  are matrices with monomial entries describing the influence of transitions on each other.

According to Theorem 1, under the earliest functioning rule, the input-output (resp. perturbation-output) transfer function matrix  $H$  (resp.  $G$ ) of the system is equal to  $CA^*B$  (resp.  $CA^*$ ).

$$y = CA^*Bu \oplus CA^*q = Hu \oplus Gq \quad (5)$$

Therefore, the condition for holding the maximization (preserving the input-output and perturbation-output behaviors) is rephrased in terms of transfer function matrices. The conditions is now to preserve the input-output and perturbation-output transfer function matrices.

When an element  $s$  of  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$  is used to code information concerning a transition of a TEG, then a monomial  $\gamma^k \delta^t$  with  $k, t \geq 0$  may be interpreted as "at most  $k$  events occur strictly before time  $t$ " (*i.e.*,  $d_s(K) \geq t$ ). An element of  $s$  of  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ , used to code a transfer relation between two transitions of a TEG (*e.g.*, an entry of  $H$ ), is causal (*i.e.*, no anticipation in the time/event domain: all exponents are non-negative) and periodic (*i.e.*,  $s = p \oplus q r^*$  with polynomials  $p, q$  and a monomial  $r \neq e$ ). For a periodic series  $s$  with  $r = \gamma^\nu \delta^\tau$ , its asymptotic slope  $\sigma(s)$  is defined as  $\nu/\tau$ .

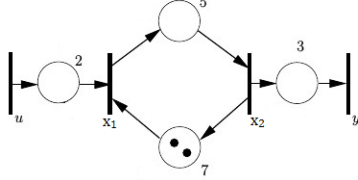


Fig. 2. TEG example

*Example 6.* Let us consider a manufacturing system depicted by the TEG given in Fig. 2. The transition labeled  $u$  represents the inputs of raw material, it is transported during 2 time units to a machine with 2 treatment spots. Its input is labelled  $x_1$ , the processing time is equal to 5, and the machine's output is labeled  $x_2$ . The processed part is then transported out of the system during 3 time units, the transition  $y$  represents when the part is out of the production line. Before accepting a new raw part, the machine must be cleaned, this operation spends 7 time units. The model is then given by

$$\tilde{A} = \begin{bmatrix} \varepsilon & \gamma^2 \delta^7 \\ \delta^5 & \varepsilon \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} \delta^2 \\ \varepsilon \end{bmatrix} \quad \tilde{C} = [\varepsilon \ \delta^3]$$

It must be noticed that this system can be realized in a straightforward way in  $(\max, +)$  or  $(\min, +)$  form:

$$\begin{aligned} x_1(k) &= \max(2 + u(k), & x_1(t) &= \min(u(t-2), \\ & 7 + x_2(k-2)) & & 2 + x_2(t-7)) \\ x_2(k) &= 5 + x_1(k) & x_2(t) &= x_1(t-5) \\ y(k) &= 3 + x_2(k), & y(t) &= x_2(t-3), \end{aligned}$$

where  $x_i(k)$  represents the firing date of part  $k$  and  $x_i(t)$  represents the number of firing occurred till the time  $t$ . These both systems are implicit equations. In order to obtain an explicit model we introduce an original procedure. We propose to split the system in the following way  $\tilde{A} = A_r \oplus A_d \oplus A_g$  where

- if  $n_{ij} \neq 0$  and  $t_{ij} \neq 0$ ,  $(A_r)_{ij} = \tilde{A}_{ij} = \gamma^{n_{ij}} \delta^{t_{ij}}$  else  $(A_r)_{ij} = \varepsilon$
- if  $t_{ij} = 0$ ,  $(A_g)_{ij} = \tilde{A}_{ij} = \gamma^{n_{ij}}$  else  $(A_g)_{ij} = \varepsilon$
- if  $n_{ij} = 0$ ,  $(A_d)_{ij} = \tilde{A}_{ij} = \delta^{t_{ij}}$  else  $(A_d)_{ij} = \varepsilon$

In the present example:

$$A_g = \begin{pmatrix} \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{pmatrix} \quad A_d = \begin{pmatrix} \varepsilon & \varepsilon \\ \delta^5 & \varepsilon \end{pmatrix} \quad A_r = \begin{pmatrix} \varepsilon & \gamma^2 \delta^7 \\ \varepsilon & \varepsilon \end{pmatrix}$$

Knowing that  $x = \tilde{A}x \oplus \tilde{B}u$  separating the matrix  $\tilde{A}$  we have:

$$\begin{aligned} x &= (A_d \oplus A_g \oplus A_r)x \oplus \tilde{B}u \\ x &= (A_d \oplus A_g)x \oplus A_r x \oplus \tilde{B}u \end{aligned}$$

Using Theorem 1

$$x = (A_d \oplus A_g)^* A_r x \oplus (A_d \oplus A_g)^* \tilde{B}u$$

Knowing that, we have  $A = (A_d \oplus A_g)^* A_r$  and  $B = (A_d \oplus A_g)^* \tilde{B}$  and these matrices generate a model in the form

$$\begin{cases} x = Ax \oplus Bu \\ y = Cx \end{cases}$$

Which can be realized in an explicit form whether in  $(\max, +)$  or  $(\min, +)$

$$(A_d \oplus A_g) = \begin{pmatrix} \varepsilon & \varepsilon \\ \delta^5 & \varepsilon \end{pmatrix} \Rightarrow (A_d \oplus A_g)^* = \begin{pmatrix} e & \varepsilon \\ \delta^5 & e \end{pmatrix}$$

$$A = (A_d \oplus A_g)^* A_r = \begin{pmatrix} \varepsilon & \gamma^2 \delta^7 \\ \varepsilon & \gamma^2 \delta^{12} \end{pmatrix}$$

and for the input

$$B = (A_d \oplus A_g)^* \tilde{B} = \begin{pmatrix} \delta^2 \\ \varepsilon \end{pmatrix},$$

this leads to the following explicit model

$$\begin{aligned} x_1(k) &= \max(2 + u(k), & x_1(k) &= \min(u(t-2), \\ & 7 + x_2(k-2)) & & 2 + x_2(t-7)) \\ x_2(k) &= \max(7 + u(k), & x_2(k) &= \min(u(t-7), \\ & 12 + x_2(k-2)) & & 2 + x_2(t-12)) \\ y(k) &= 3 + x_2(k) & y(k) &= x_2(t-3) \end{aligned}$$

These matrices  $A$  and  $B$  can be programmed into a software without realization problems. The system can be solved by considering Theorem 1, (see Eq. (5)) and the transfer matrix  $y = Hu$  is given by  $H = \tilde{C} \tilde{A}^* \tilde{B} = \tilde{C} A^* B = \delta^{10} (\gamma^2 \delta^{12})^*$ . This computation can be easily performed by using the library MinMaxgd available as a C++ library or alternatively thanks to a web interface (Cottenceau et al. (2006); Cândido et al. (2017)).

#### 4. OBSERVER BASED CONTROLLER

This section presents how to implement an efficient control strategy for dynamical systems considered in the previous section. The control strategy proposed is depicted in Fig. 3. It is inspired from the observer based control for classical linear systems (Hardouin et al., 2018, 2010).

The motivation to control the input of these systems is to decide when the operator should start to achieve an

objective, e.g. when do you start the departure of a processing operator in order to achieve the customer demand. Hence, the aim is to design a controller able to decide when the system should start to work in order to achieve a desired behavior. Classically, a popular production policy is to design a just-in-time policy, that is, to start as late as possible while ensuring the customer demand. It minimizes the internal stock while keeping the performance.

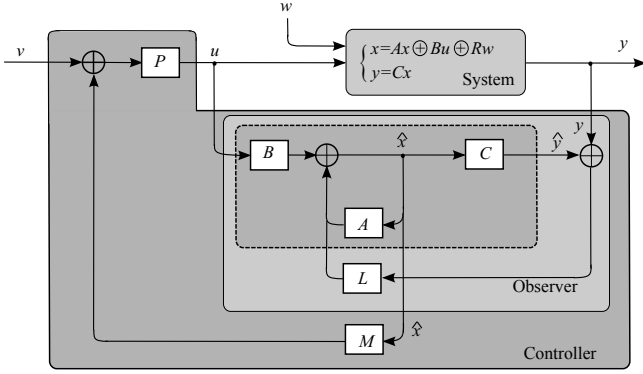


Fig. 3. Observer-based Controller architecture

The design goal is to get controllers  $M$  and  $P$  (see Fig. 3) which are matrices ensuring that the control input  $u = P(v \oplus M\hat{x})$  be the greatest (*i.e.* the one which delays as much as possible the input) in order to achieve a given objective, the reference input  $v$ . Signal  $\hat{x}$  is either the real state of the system, ( $\hat{x} = x$  if the state is measurable), or an estimation  $\hat{x}$  observed thanks to an observer, inspired from the Luenberger observer (Luenberger (1971)). This estimator is given by

$$\hat{x} = A\hat{x} \oplus Bu \oplus L(\hat{y} \oplus y). \quad (6)$$

Where  $L$  is an observer matrix to be designed. It is fed by the measured output  $y$  of the system and ensures that the real system output be possible to compute the estimator  $\hat{x}$ , especially that disturbance  $w$  feeding the system through matrix  $R$ . This observer based controller is then a feedback control strategy. The goal is to design  $P, M, L$  in order to achieve a desired behavior denoted  $G_{ref}$ .

By solving Eq. 6  $\hat{x}$  is given by

$$\begin{aligned} \hat{x} &= A\hat{x} \oplus Bu \oplus L(Cx \oplus C\hat{x}) \\ &= (A \oplus LC)^*Bu \oplus (A \oplus LC)^*LCA^*Rw \end{aligned}$$

by repeating in  $u$ , the control is:

$$\begin{aligned} u &= P(v \oplus M\hat{x}) \\ &= P(M(A \oplus LC)^*BP)^*v \\ &\quad \oplus PM((A \oplus LC)^*BPM)^*(A \oplus LC)^*LCA^*Rw \end{aligned}$$

The development are given in (Hardouin et al. (2010)) and lead to the optimal design

$$P_{opt} = (CA^*B) \setminus G_{ref} \quad (7)$$

$$L_{opt} = ((A^*B) \setminus (CA^*B)) \wedge ((A^*R) \setminus (CA^*R)) \quad (8)$$

$$M_{opt} = P_{opt} \setminus P_{opt} \setminus (A^*BP_{opt}) \quad (9)$$

#### 4.1 Example

In our sample system, showed in Fig. 2, since it does not have many inner states or inputs and outputs, these calculations can be done by hand and drawn. Using  $G_{ref} = CA^*B$ , we are able to calculate  $P_{opt} = (CA^*B) \setminus (CA^*B)$  knowing that  $(CA^*B) = \delta^{10}(\gamma^2\delta^{12})^*$  we obtain

$$P_{opt} = \delta^{10}(\gamma^2\delta^{12})^* \setminus \delta^{10}(\gamma^2\delta^{12})^* = (\gamma^2\delta^{12})^*$$

the equality holds since  $a^* \setminus a^* = a^*$  (see Baccelli et al. (1992)) which yields that  $(\gamma^2\delta^{12})^* \setminus (\gamma^2\delta^{12})^* = (\gamma^2\delta^{12})^*$ .

With the result of  $P_{opt}$  we are able to calculate  $M_{opt}$

$$M_{opt} = (\gamma^2\delta^{12})^* \setminus (\gamma^2\delta^{12})^* \setminus \left( \frac{\delta^2}{\delta^7} \right) \otimes (\gamma^2\delta^{12})^*$$

$$M_{opt} = (\delta^{-2} \delta^{-7}) \otimes (\gamma^2\delta^{12})^*$$

Obviously it is not possible to implement a controller that has negative exponents (this controller would be non-causal), hence the solution is to pick only the causal projection. To do this, imagine a Cartesian plane where gamma is the  $x$  axis and delta the  $y$  axis. Now we put the points according to the desired series, for example  $\gamma^{-4}\delta^{-1} \oplus \gamma^{-2}\delta^2 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^4$ .

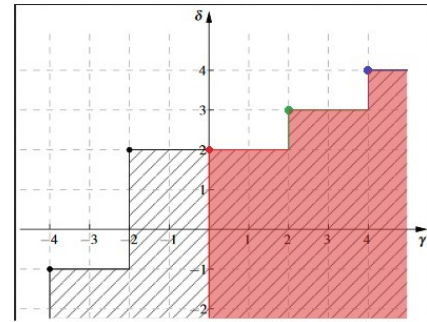


Fig. 4. Gamma-delta plane representation of series  $\gamma^{-4}\delta^{-1} \oplus \gamma^{-2}\delta^2 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^4$  and its causal projection  $\delta^2 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^4$ .

It is clear that, using the south-east cone presented in section 2 all the points in the drawing must be inside the north-east quadrant for the series to be realizable. This way, the causal projection is the biggest area possible, in a way that all its corners are inside this quadrant, and is contained in the original area. For our example, it would be  $\delta^2 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^4$ , as showed in Fig. 4

Using the same reasoning in  $M_{opt}$  we get the following causal series:

$$Pr_+(M_{opt}) = (\gamma^2\delta^{10} \gamma^2\delta^5) \otimes (\gamma^2\delta^{12})^*$$

Finally the observer needs to be calculated

$$L_1 = \begin{pmatrix} \delta^2 \\ \delta^7 \end{pmatrix} \otimes (\gamma^2 \delta^{12})^* \not\prec \delta^{10} \otimes (\gamma^2 \delta^{12})^* = \begin{pmatrix} \delta^{-8} \\ \delta^{-3} \end{pmatrix} \otimes (\gamma^2 \delta^{12})^*$$

$$L_2 = \begin{pmatrix} e & \gamma^2 \delta^7 \\ \delta^5 & e \end{pmatrix} \otimes (\gamma^2 \delta^{12})^* \not\prec \begin{pmatrix} \delta^8 \\ \delta^3 \end{pmatrix} \otimes (\gamma^2 \delta^{12})^* = \begin{pmatrix} \gamma^2 \delta^4 \\ \delta^{-3} \end{pmatrix} \otimes (\gamma^2 \delta^{12})^*$$

Then  $L = L_1 \wedge L_2$  :

$$L = \begin{pmatrix} \gamma^2 \delta^4 \\ \delta^{-3} \end{pmatrix} \otimes (\gamma^2 \delta^{12})^* \Rightarrow Pr_+(L) = \begin{pmatrix} \gamma^2 \delta^4 \\ \gamma^2 \delta^9 \end{pmatrix} \otimes (\gamma^2 \delta^{12})^*.$$

Since all the matrices entries are causal, all these controllers are realizable, and thus they can be implemented.

To convert the gamma-delta elements into min-plus or max-plus equations, we propose to split the series  $s = p \oplus qr^*$  in the following way : first we define  $\zeta_k = qr^*$  thus  $s = p \oplus \zeta_k$ . By using Theorem 1, we get the relation  $\zeta_k = r\zeta_k \oplus q$ . In our example  $\hat{x} = A\hat{x} \oplus Bu \oplus Ly$ , for example sake, let us focusing on the first state we obtain,  $\hat{x}_1 = \gamma^2 \delta^7 x_2 \oplus \delta^2 u \oplus \gamma^2 \delta^4 \otimes (\gamma^2 \delta^{12})^* y$ . Hence in this case  $p = \gamma^2 \delta^7 x_2 \oplus \delta^2 u$ ,  $q = \gamma^2 \delta^4 y$  and  $r = \gamma^2 \delta^{12}$ . By using the formula we just found,  $\hat{x}_1 = \gamma^2 \delta^7 x_2 \oplus \delta^2 u \oplus \zeta_1$  and  $\zeta_1 = \gamma^2 \delta^{12} \zeta_1 \oplus \gamma^2 \delta^4 y$ . Thanks to this expression it is straightforward to put these equations in min-plus or max-plus implicit equations. For simplicity, min-plus will be used because updating the system periodically, in our case each 500ms, is easier than when events occur. The min-plus equation for  $\hat{x}_1$  is  $\hat{x}_1 = \min(2 + \hat{x}_2(t-7), u(t-2), \zeta_1(t))$  knowing that  $\zeta_1(t) = \min(2 + \zeta_1(t-12), 2 + y(t-4))$ . These equations can be easily programmed into a software, since they only require data storage to use the time delays properly.

## 5. REAL SYSTEM APPLICATION

In this section, we are interested in applying this method in the real system, depicted in Fig. 5, which is located in Polytech Angers, France.

The system has 2 separated sections, a faster loop and a slower loop as shown in Fig. 5. The slower loop has 6 buttons that do not let the pallets pass while the faster loop has only 4. All the buttons have sensors just before them, as it can be seen in Fig. 6. The pallet's size is such that if they are waiting for the button, the sensor will stay active. Each section (between two buttons) has a defined maximum number of pallets. The travel times were measured 10 times, and the time used is the average of them.



Fig. 5. Slower Loop

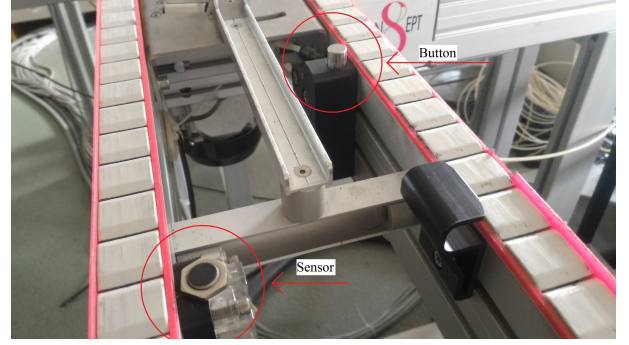


Fig. 6. Button and Sensor

Each button (labelled B1 to B10), the travel time and the maximum number of pallets are represented in Fig. 7. It is important to notice that there are 3 pallets waiting for B1, 2 waiting for B5 and 1 waiting for B6, as initial conditions of the system.

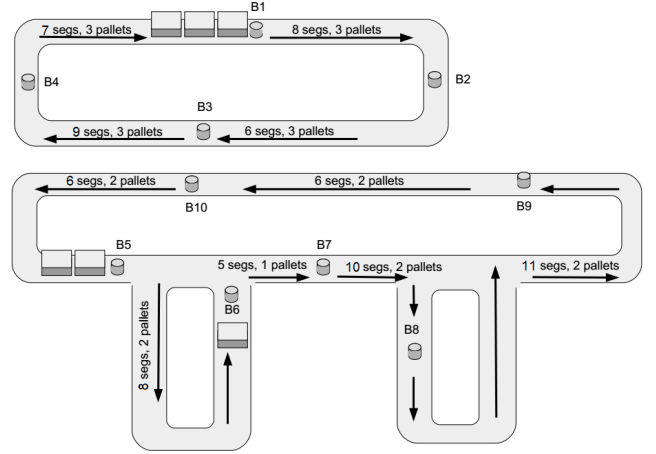


Fig. 7. Buttons, travel times (in sec) and pallet limit

The system is programmed to activate the buttons when there is one pallet waiting for it (the sensor is active), there is at least one space left for the path and at least one control token available. Especially for B3 and B10, there is a forced synchronization, meaning that B3 and B10 will always activate at the same time, requiring 2 control tokens, one free space between B10 and B5 and

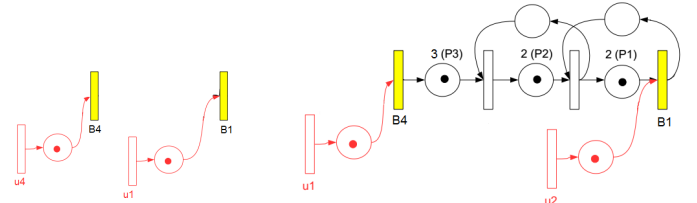


Fig. 8. Modeling 1

Fig. 9. Modeling 2

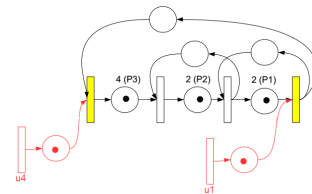


Fig. 10. Final path Model

1 pallet waiting for B3 and another waiting for B10. It is worth mentioning that after each consecutive activation of any button, it will wait for 2 seconds until it activates again.

Now the first step is to put this system in a Petri net model. We will analyze the section between B4 and B1 and then apply the same reasoning to the other sections. Since each button changes the state of the system, we have the first two inner states, each one with one associated input because of the control tokens as shown in Fig. 8. Then we will need one place for each empty slot in the section, and since 2 pallets cannot be in the same place, these places must have only one token at a time. Besides the tokens, the timings for each place has to be 2 seconds, since the button will only activate 2 seconds after its previous activation, but the sum of all the places between the buttons has to be the total travelling time, using these conditions we have the second step of the model in Fig. 9. It is worth mentioning that the initial pallets are the tokens into (P1), (P2) and (P3) so the token in (P1) can activate B1 instantaneously.

To summarize the model so far, the token in P1 will activate the B1 transition, meaning that the button will go down and the pallet will enter the path between B1 and B2. After that, the two other pallets will begin moving and wait for the 2 seconds to pass, what explains the 2 seconds travelling time in the model. The last condition that is missing is the total number of pallets for each section. Since for the section B4 to B1 only 3 pallets are allowed, and all the three are already in it, so we have the final model represented in Fig. 10.

If we replicate this reasoning to all sections combined and knowing that before every button we have a sensor indicating the pallet is there, this means the sensors are the outputs of the system, we get the final model represented in Fig. 11.

We can achieve the system's matrices as proposed in section 3.1, where  $A_0$  will be split into  $A_g, A_d$  and  $A_r$  to make the model implementable using the same procedure discussed in section 3. The next step is to use these 3 matrices to calculate  $P_{opt}, L_{opt}$  and  $M_{opt}$  thanks to Eqs. (7-9). All the calculus needed to achieve the final controller were obtained by using the software MinmaxGD (Cottenceau et al., 2006). After executing, this software gave us the matrices just like the ones presented in section 4.1.

Transforming those matrices in code is extremely laborious and can lead to unexpected errors. Knowing that, a software has been developed to make the calculus and write a C code directly (see Freitas and Hardouin (2017)), using the method shown at the end of section 4.1. The generated code was then executed into a PC that supervises the real system. The PLCs ensure that the system will comport as specified and provide data for the supervisory system. The code works in a way that it reads the outputs provided by the PLCs, updates the input control tokens consequently updating the input and thus updating the amount of control tokens available for the buttons, then updates the observer for the user to know if the system is comporting adequately every 500 ms.

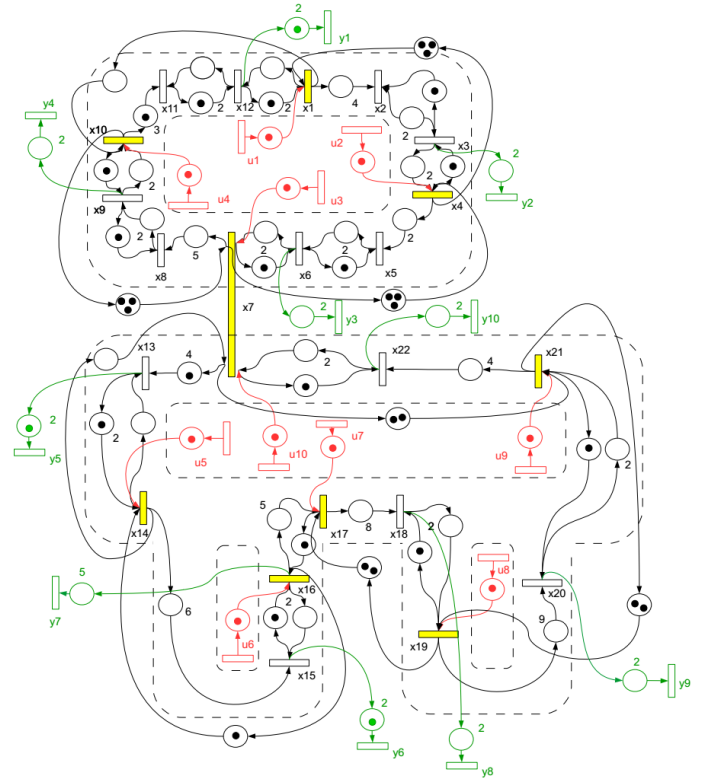


Fig. 11. Complete Model (Gonçalves, 2015)

## 6. RESULTS

In this section we are going to discuss the effects of having implemented the control into the real system. Since the only place that can occur pallet accumulation is in B4 and B10 and the loop containing B4 is faster, it is the only place that can have accumulation. Our chosen  $G_{ref}$  was the system's transfer function, that means, we chose to maintain the original output, but to delay as much as possible the buttons activations in order to reduce stocking. Knowing that, the objective of the controller is to reduce the stocking in B4 without changing the original output. To illustrate the performance of the controller we focus on measurements of  $x_7$  state.

It must be mentioned that the travelling times are not deterministic, hence we focus on the difference between  $x_7$  state with and without control, sometimes this difference is positive and sometimes negative. In Fig. 12 it is possible to see the time evolution of this difference. As expected, the difference is insignificant principally because  $x_7$  is the number of times the state has been activated, which is a number that increases over time. The mean value of the difference is -0.1342, very close to 0, showing that all outputs would have been maintained if the system was perfectly deterministic.

The next topic will be the stocking difference. For this matter the PLCs is programmed to measure  $x_4$ 's information, this way the stock is calculated using  $stock(t) = x_4(t) - x_7(t)$ . The result can be seen in Fig. 13.

This pattern keeps repeating for the whole simulation. As it is possible to see, the controlled system has at maximum 2 pallets in stock, while the not controlled (natural) system



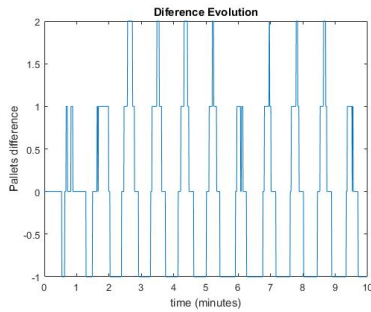


Fig. 12. Difference between  $x_7$  with (Natural) and without controller (Controlled)

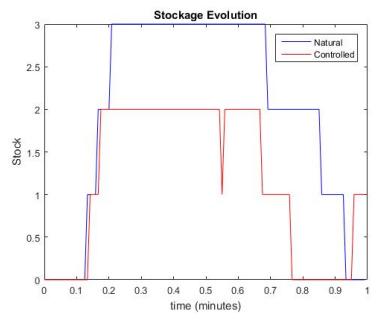


Fig. 13. Stock Evolution with and without control

has 3. This result makes sense, since our aim is to maintain the outputs, in this case, for output  $y_2$  to be maintained, the system has to activate B2 at least twice, resulting in a minimum stock of 2. Calculating the mean stock, we get 1.6550 pallets for the natural system, and 1.3175 for the controlled one, meaning a 20.4% stock reduction.

## 7. CONCLUSION

In this paper we propose and present a way to implement in an efficient way controllers for max-plus linear systems. Freely available software tools yield automatically the code for implementation in a SCADA system. In the real example presented, the just-in-time control implemented has reduced the useless internal stocks by 20%, which illustrates the interest of this kind of production policy. The engineers can use this methodology to adapt the control to their specific systems in an easy way. A user-friendly interface will be develop soon in order to help them to select the control strategy they want to implement among the different available possibilities (Observer-based control (Hardouin et al., 2018), Disturbance Decoupling Problem (Shang et al., 2016), Robust control (Lhommeau et al., 2005), Stabilization thanks to Feedback Control (Cottenceau et al., 2001), Model Predictive Control (Schutter and Boom, 2001)).

## REFERENCES

Baccelli, F., Cohen, G., Olsder, G. J., Quadrat, J. P., 1992. Synchronization and Linearity : An Algebra for Discrete Event Systems. Wiley and Sons.  
 Cândido, R. M. F., Lhommeau, M., Hardouin, L., Santos-Mendes, R., 2017. Minmaxgdjs : A web toolbox to handle periodic series in minmax[[gamma,delta]] semiring. In: IFAC World Congress. Toulouse, <http://perso-laris.univ-angers.fr/~lhommeau/>.

Cohen, G., Gaubert, S., Quadrat, J.-P., 1998. Max-plus algebra and system theory : Where we are and where to go now. In: IFAC Conference on System Structure and Control. Nantes.  
 Cottenceau, B., Hardouin, L., Boimond, J.-L., 2001. On timed event graphs stabilization by output feedback in dioid. In: 1st IFAC Symposium on System Structure and Control, Workshop on (max,+) algebras. Prague.  
 Cottenceau, B., Hardouin, L., Lhommeau, M., 2006. Minmaxgd a library for computation in semiring of formal series [http://perso-laris.univ-angers.fr/~hardouin/documentation\\_minmaxgd.pdf](http://perso-laris.univ-angers.fr/~hardouin/documentation_minmaxgd.pdf).  
 Freitas, G., Hardouin, L., 2017. State estimation of an automated system in max-plus algebra by using a linear observer <http://lisabiblio.univ-angers.fr/MASTERSiteWebFreitasOliveiraGabriel/Home.html>.  
 Gonçalves, V. M., 2015. Tropical algorithms for linear algebra and linear event-invariant dynamical systems. Ph.D. thesis, Advisors C. A. Maia and L. Hardouin, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, <http://perso-laris.univ-angers.fr/~hardouin/ThesisViniciusMariano.pdf>.  
 Hardouin, L., Cottenceau, B., Shang, Y., Raisch, J., 2018. Control and state estimation for max-plus linear systems. Foundations and Trends in Systems and Control 6 (1), 1–116.  
 Hardouin, L., Maia, C. A., Cottenceau, B., Santos-Mendes, R., Sep. 2010. Max-plus Linear Observer: Application to manufacturing Systems. In: Proceedings of the 10th International Workshop on Discrete Event Systems, WODES'10. Berlin, pp. 171–176.  
 Heidergott, B., Olsder, G., van der Woude, J., 2006. Max Plus at Work: Modeling and Analysis of Synchronized Systems : a Course on Max-Plus Algebra and Its Applications. Princeton University Press, <https://books.google.fr/books?id=U9uT1JV6sTcC>.  
 Lhommeau, M., Hardouin, L., Ferrier, J.-L., Ouerghi, I., Dec. 2005. Interval analysis in dioid : Application to robust open loop control for timed event graphs. In: Proceedings of the 44th IEEE CDC-ECC Conference, pp. 7744–7749.  
 Luenberger, D. G., 1971. An Introduction to observers. IEEE Trans. on Automatic Control 16 (6), 596–602.  
 Maia, C. A., Hardouin, L., Santos Mendes, R., Cottenceau, B., Dec. 2005. On the Model Reference Control for Max-Plus Linear Systems. In: Proceedings of the 44th CDC-ECC Conference, pp. 7799–7803.  
 Schutter, B. D., Boom, T. v. d., Jul. 2001. Model predictive control for max-plus-linear discrete event systems. Automatica vol. 37 (7).  
 Schutter, B. D., van den Boom, T., Xu, J., Farahani, S., 2019. Analysis and control of max-plus linear discrete-event systems: An introduction. Discrete Event Dynamic Systems.  
 Shang, Y., Hardouin, L., Lhommeau, M., Maia, C. A., 01 2016. An integrated control strategy to solve the disturbance decoupling problem for max-plus linear systems with applications to a high throughput screening system. Automatica 63, 338–348.