Master Systèmes Dynamiques et Signaux

Rapport Bibliographique

State estimation of an automated system in max-plus algebra by using a linear observer

---

Auteur :
M. Gabriel FREITAS OLIVEIRA

Encadrant :
M. Laurent HARDOUIN

Jury :
Dr. Laurent HARDOUIN
Dr. Mehdi LHOMMEAU
Dr. Remy GUYONNEAU
Dr. Sebastien LAGRANGE

Version du
July 1, 2017

Acknowledgements

I would like to say thank you to CAPES, that funded my stay and research in Angers, Professor Laurent Hardouin, that gave all of us a warm welcome and is always in good mood to answer our questions. A big thank you to the Brazilian professors Carlos Andrey Maia, who guided me and selected me to this exchange and Rafael Santos Mendes, the project's coordinator that welcomed us in Angers, and gave all of us many advices for our stay.

Contents

List of Figures

List of acronyms

TEG *Timed Event Graph*

Introduction

The industrial manufacturing systems can be modeled using a subclass of Petri Nets named Timed Event Graphs (TEGs), that is a timed discrete event system subject to synchronization and delay phenomena. The synchronization represents a meeting between events, and the delay corresponds to transportation times. A specific control theory has been developed during the last decades [1, 2, 3, 4, 5, 6]. This work aims to purpose a specific software tools to implement automatically an observer-based controller on a real system. It is decomposed on several steps first the model of the system is given as a TEG, then the corresponding state-model of this system is given to the software as an input. A specification representing the desired behavior is also given as an input. Then the software will give automatically the code to implement the control law in a Supervisory Control and Data Acquisition (SCADA) system.

This work is organized as follows: First, the algebraic tools necessary to synthesize the control law are recalled, then in Chapter 2 the modeling method of a system is presented. The proposed control laws given in Chapter 3 is an observer-based control strategy which aims to match a reference model. Later an illustration example is presented in Chapter 3. Finally a discussion on the reference model choice is proposed in Chapter 4.

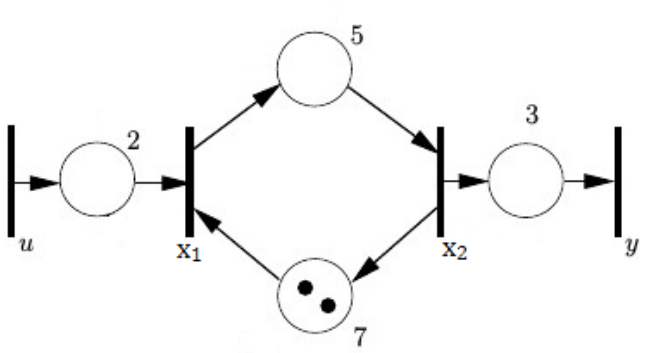

Figure 1: TEG example

Controlling a discrete event system in an industrial space is desirable, many times using the just-in-time criteria, this means, the output will be smaller than a reference output,

but the input will be delayed as long as possible. This avoids accumulation in machines or lines, for example, a raw material will be the less amount of time out of the fridge. For example sake, this paper will use the system in figure 1 to demonstrate the TEGs and the algebra used.

Chapter 1

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 ε and e respectively, and ε 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 $a \succeq b \Leftrightarrow a = a \oplus b \Leftrightarrow \ b = a \wedge b$ (where $a \wedge b$ is the greatest lower bound of a and b), therefore an idempotent semiring $\mathcal{S}$ is a partially ordered set (see [7, 4] 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 [7], 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 $f : \mathcal{D} \to \mathcal{E}$, 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 ([7], 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.*,

$$(A \otimes B)_{ik} = \bigoplus_{j=1\ldots n} (a_{ij} \otimes b_{jk}) \tag{1.1}$$

$$(A \oplus B)_{ij} = a_{ij} \oplus b_{ij}, \tag{1.2}$$

$$(A\backslash B)_{ij} = \bigwedge_{k=1..n} (a_{ki}\backslash b_{kj}) \ , \ (B/A)_{ij} = \bigwedge_{k=1..n} (b_{ik}/a_{jk}). \tag{1.3}$$

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

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 $\leqslant$. 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 $\varepsilon \in \overline{\mathbb{Z}}_{\max}$ 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 $\overline{\mathbb{Z}}_{\max}$,

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

we get

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

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

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

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

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 in case of scalar multiplication in max-plus algebra, while this is not true for the matrix case.

Chapter 2

System Modeling

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

Dioid $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ is formally the quotient dioid of $\mathbb{B}[\![\gamma, \delta]\!]$ (the set of formal power series in two commutative variables γ and δ, with Boolean coefficients ant 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 (γ, δ)-plane. Consider a monomial $\gamma^k\delta^t \in \mathbb{B}[\![\gamma, \delta]\!]$, its south-east cone is defined as $\{(k', t')|k' \geqslant k \text{ and } t' \leqslant t\}$. 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 $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$

$$\gamma^k \oplus \gamma^t = \gamma^{min(k,t)} \text{ and } \delta^k \oplus \delta^t = \delta^{max(k,t)} \tag{2.1}$$

- a simple formulation of the order relation for monomials

$$\gamma^n\delta^t \preceq \gamma^{n'}\delta t' \Leftrightarrow n \geqslant n' \text{ and } t \leqslant t'$$

A simple interpretation of the variable γ and δ for daters is available:

- multiplying a series s by γ is equivalent to shifting the argument of the associated dater function by -1.

- multiplying a series s by δ 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 figure 2.1. 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 of 2.1.

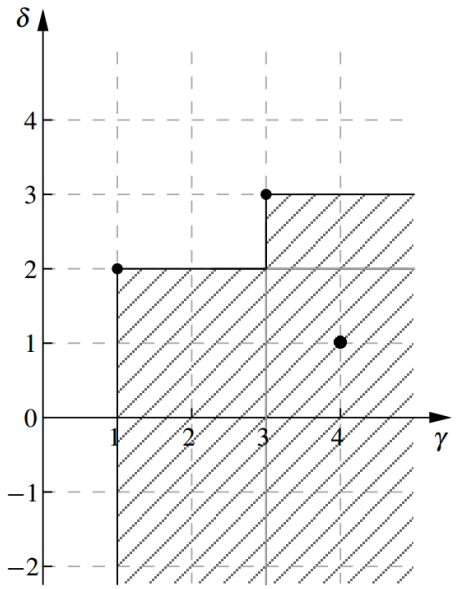

Figure 2.1: s and its south-east cone (hatched)

Besides,

$$s = \bigoplus_{k\leqslant 0} \gamma^k\delta^{-\infty} \oplus \bigoplus_{k=1,2} \gamma^k\delta \oplus \bigoplus_{k\geqslant 3} \gamma^k\delta^4$$

Therefore, the dater d_s associated with s is given by

$$d_s(k) = \begin{cases} -\infty & \text{if } k \leqslant 0 \\ 1 & \text{if } k = 1,2 \\ 4 & \text{if } k \geqslant 2 \end{cases}$$

Knowing this, the following relations are presented

$$\gamma^n\delta^t \oplus \gamma^{n'}\delta^t = \gamma^{min(n,n')}\delta^t \tag{2.2}$$
$$\gamma^n\delta^t \oplus \gamma^n\delta^{t'} = \gamma^n\delta^{max(t,t')} \tag{2.3}$$
$$\gamma^n\delta^t \wedge \gamma^{n'}\delta^{t'} = \gamma^{max(n,n')}\delta^{min(t,t')} \tag{2.4}$$
$$\gamma^n\delta^t \otimes \gamma^{n'}\delta^{t'} = \gamma^{n+n'}\delta^{t+t'} \tag{2.5}$$
$$\gamma^n\delta^t \not/ \gamma^{n'}\delta^{t'} = \gamma^{n-n'}\delta^{t-t'} \tag{2.6}$$
$$\gamma^n\delta^t \backslash\!\!\!\not\, \gamma^{n'}\delta^{t'} = \gamma^{n'-n}\delta^{t'-t} \tag{2.7}$$

2.2 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, γ is a shift operator in the event domain, where an event is interpreted as the firing of the transition, and δ is a shift operator in the time domain.

The transitions of a TEG are divided into three categories:

- state transitions $(x_1, ..., x_n)$: transitions with at least one input place and one output place.
- input transitions $(u_1, ...u_p)$: transitions with at least one output place, but no input places.
- output transitions $(y_1, ...y_m)$: 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 x_i and downstream from transition x_j, the influence of transition x_j on transition x_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 τ_{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 controllable input, $y \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^m$ the output and $w \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^n$ the uncontrollable of the state. The uncontrollable input w delays firing of internal transition, it can model for example unexpected failure, delays or uncertain parameters such as task duration, while matrix $R \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{n\times n}$ depicts how these perturbations affect the inner states. If each transition can be affected by disturbance, R is the Identity matrix. $A \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{n\times n}$, $B \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{n\times p}$, $C \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{m\times n}$ 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^*).

$$y = CA^*Bu \oplus CA^*Rw = Hu \oplus Gw \tag{2.8}$$

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 $k,t \geqslant 0$ may be interpreted as "at most k events occur strictly before time t" (*i.e.* , $d_s(K) \geqslant t$). 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.* , $s = p \oplus qr^*$ 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 ν/τ.

Example 6

Let us consider a manufacturing system depicted by the TEG given in Figure 1. The transition labeled u represents the inputs of raw material, it is transported during 2 seconds to a machine with 2 treatment spots. Its input is labeled 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 times unit, 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 times unit.

To put this system into gamma-delta equations we use the time delay as delta exponent and the amount of initial tokens as gamma exponent for each arrow entering the transition, e.g $x_1 = \gamma^2\delta^7 x_2 \oplus \delta^2 u$. The complete model is then given by

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

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

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

Where $x_i(k)$ represents the firing date of part k.

Where $x_i(t)$ represents the number of firing occurred till the time t.

These both systems are implicit equations, in order to obtain an explicit model we propose to split the system in the following way $\widetilde{A} = A_r \oplus A_d \oplus A_g$ where

- if $n_{ij} \neq 0$ and $t_{ij} \neq 0$, $(A_r)_{ij} = \widetilde{A}_{ij} = \gamma^{n_{ij}}\delta^{t_{ij}}$ else $(A_r)_{ij} = \varepsilon$
- if $t_{ij} = 0$, $(A_g)_{ij} = \widetilde{A}_{ij} = \gamma^{n_{ij}}$ else $(A_g)_{ij} = \varepsilon$
- if $n_{ij} = 0$, $(A_d)_{ij} = \widetilde{A}_{ij} = \delta^{t_{ij}}$ 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 $x = \widetilde{A}x \oplus \widetilde{B}u$ separating the matrix $\widetilde{A}$ we have:

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

Using Theorem 1

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

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

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

Which can be realized in an explicit form either 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

$$B = (A_d \oplus A_g)^* \widetilde{B} = \begin{pmatrix} \delta^2 \\ \delta^7 \end{pmatrix}$$

Which generate the explicit model firstly for (max, +)

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

and for (min,+)

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

These matrices A and B can be programmed into a software without realization problems. The system can be solved by considering Theorem 1, (see equation 2.8) and the transfer matrix $y = Hu$ is given by $H = \widetilde{C}\widetilde{A}^*\widetilde{B} = CA^*B = \delta^{10}(\gamma^2\delta^{12})^*$. This computation can be easily be performed by using the library MinMaxgd available as a C++ library as web interface see [8]

Chapter 3

Observer Based Controller

This chapter presents how to implement an efficient control strategy for dynamical systems considered in the previous chapter. The control strategy proposed is depicted in figure 3.1. It is inspired from the observer based control for classical linear systems.

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 aims 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 performances.

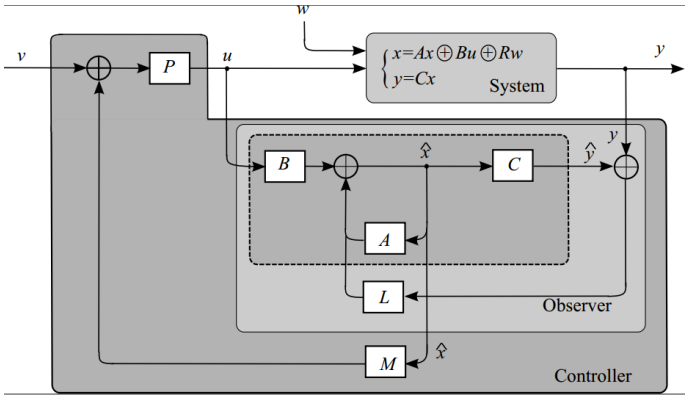

Figure 3.1: Controlled system

The design goal is then, to get controller M and P (see 3.1) which are matrices ensuring that the control input $u = P(v \oplus M\hat{x})$ be the greatest (*i.e.* the one which delay 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) an estimation

$\hat{x}$ observed thanks to an observer, inspired from the Luenberger observer ([9]) for classical linear systems. This estimator is given by

$$\hat{x} = A\hat{x} \oplus Bu \oplus L(\hat{y} \oplus y) \tag{3.1}$$

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 though 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 equation 3.1 $\hat{x}$ 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:

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

The development are given in [10] and leads to the optimal design

$$P_{opt} = (CA^*B) \backslash G_{ref} \tag{3.2}$$
$$L_{opt} = ((A^*B) / (CA^*B)) \wedge ((A^*R) / (CA^*R)) \tag{3.3}$$
$$M_{opt} = P_{opt} \backslash P_{opt} / (A^* B P_{opt}) \tag{3.4}$$

3.1 Example

Since our sample system showed in Figure 1 does not have many inner states or inputs and outputs, these calculations can be done easily by hand. For this example we are going to use $G_{ref} = H$, meaning that we want the system to maintain the outputs, but delay as long as possible the inputs. In this way we are able to calculate $P_{opt} = (CA^*B) \backslash (CA^*B)$

knowing that $(CA^*B) = \delta^{10}(\gamma^2\delta^{12})^*$. Using relation 2.7

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

Since $a^* \backslash\!\!\backslash a^* = a^*$

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

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

$$M_{opt} = (\gamma^2\delta^{12})^* \backslash\!\!\backslash (\gamma^2\delta^{12})^* \not/ \begin{pmatrix} \delta^2 \\ \delta^7 \end{pmatrix} \otimes (\gamma^2\delta^{12})^*$$
$$M_{opt} = \begin{pmatrix} \delta^{-2} & \delta^{-7} \end{pmatrix} \otimes (\gamma^2\delta^{12})^*$$

It is impossible to implement a controller that has negative exponents because this controller would be non-causal. 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. Now we put the points according to the series desired, for example $\gamma^{-4}\delta^{-1} \oplus \gamma^{-2}\delta^2 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^4$.

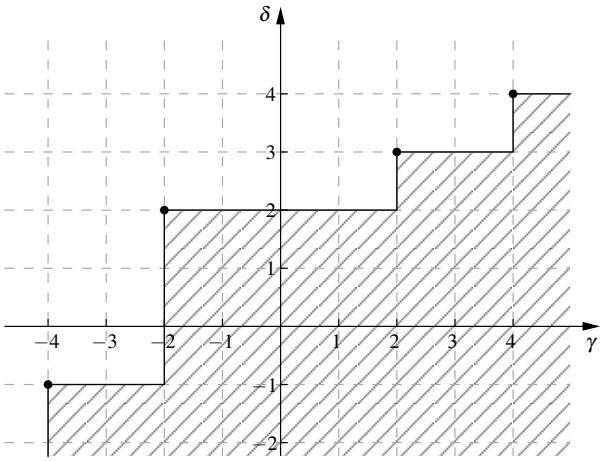

Figure 3.2: Gamma-delta plane representation

It is clear that using the south-east cone presented in Chapter 1 all the points in the drawing must be inside the first 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 the first 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 figure 3.3

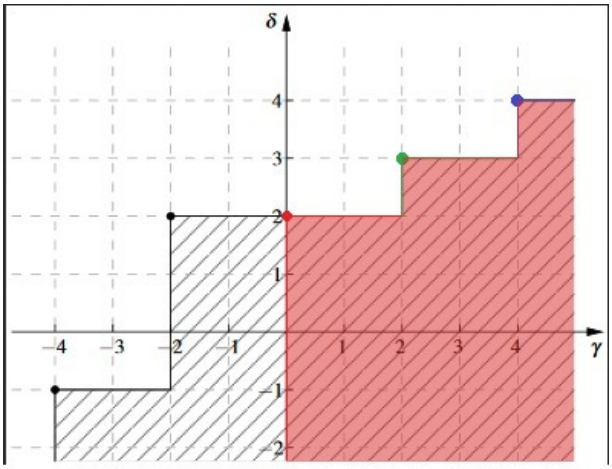

Figure 3.3: Causal series

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

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

Finally the observer needs to be calculated

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

$$L_1 = \begin{pmatrix} \delta^{-8} \\ \delta^{-3} \end{pmatrix} \otimes (\gamma^2\delta^{12})^*$$

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

$$\delta^{10} \otimes (\gamma^2\delta^{12})^* \qquad L_2 = \begin{pmatrix} \gamma^2\delta^4 \\ \delta^{-3} \end{pmatrix} \otimes (\gamma^2\delta^{12})^*$$

Now doing $L = L_1 \wedge L_2$:

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

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 will generalize the element to $S = p \oplus qr^*$ this way we define $\zeta_k = r^*$ thus $S = p \oplus q\zeta_k$. Using Theorem 1, we get the relation $\zeta_k = r\zeta_k$. In our example $\hat{x} = A\hat{x} \oplus Bu \oplus Ly$, using only the first state for example sake, $\hat{x}_1 = \gamma^2\delta^7 x_2 \oplus \delta^2 u \oplus \gamma^2\delta^4 \otimes (\gamma^2\delta^{12})^* y$. Evidently in this case $p = \gamma^2\delta^7 x_2 \oplus \delta^2 u$, $q = \gamma^2\delta^4 y$ and $r = \gamma^2\delta^{12}$. Using the formula we just found, $\hat{x_1} = \gamma^2\delta^7 x_2 \oplus \delta^2 u \oplus \gamma^2\delta^4\zeta_1$ and $\zeta_1 = \gamma^2\delta^{12}\zeta_1$. In Figure 3.4 we can see the full controlled system. It is worth mentioning that the method presented above was used for the controller, while for the observer we decided to define $\zeta_k = qr^*$ instead of $\zeta_k = r^*$, this way $\zeta_k = r\zeta_k \oplus q$, showing that we have multiple definitions that work the same.

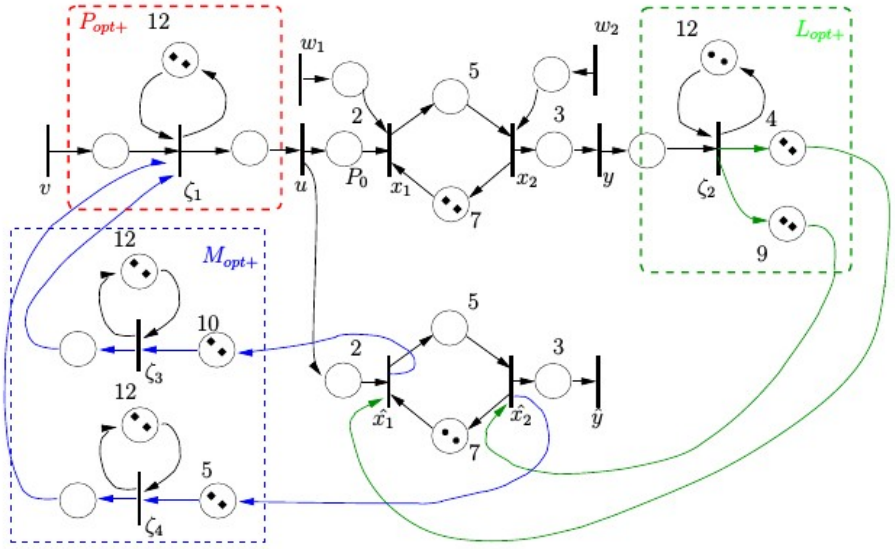

Figure 3.4: Controlled Sample System

This way we are able to put these equations in min-plus or max-plus easily. For simplicity sake, min-plus will be used because updating the system periodically 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.

All this calculus as well as the translation from gamma-delta equations into code are done automatically by a software made in the university. First, one needs to put the matrices of the modeled system $\widetilde{A}$, $\widetilde{B}$ and $\widetilde{C}$, as showed in figure 3.5.

Then, the software will calculate the observer using the relations $\hat{x} = A\hat{x} \oplus Bu \oplus Ly$ and $u = P(v \oplus M\hat{x})$. Considering $v = \varepsilon$. Substituting the first relation into the second one we get the input relation $u = PMA\hat{x} \oplus PMBu \oplus PMLy$. Next, using the same method presented in chapter 2, the realizable matrix is calculated for the input, and then finally the code is made using these results. The controller is separated in several libraries,

```
int main()
{
    smatrix A(2,2); // State Matrix
    smatrix B(2,1); // Input Matrix
    smatrix C(1,2); // Output matrix

    A(0,1) = gd(2,7); //Insert gamma^2*delta^7
    A(1,0) = gd(0,5);

    B(0,0) = gd(0,2);

    C(0,1) = gd(0,3);

    GetController(A, B, C); //Get controller code


    return 0;
}
```

Figure 3.5: Code of the example made

each one for each necessary update, for example, we need to update the auxiliary states before updating the observer. The function used to make the code also can receive a 4th parameter, allowing the user to put the perturbation matrix as a pre-defined one, in this case the default value was used as the identity matrix. If the user choses to test a controller known a priori the function $void\ GenerateCode(A, B, L, M, P)$ can be used to the only the translation from gamma delta matrices to C code.

The delays were made using circular arrays to store the information, using the smallest number of positions possible. A circular array is one such its first and last position are connected, as shown in figure 3.6, its access is made using the mod function. In this way, the values are stored into the positions, and once they are no longer useful (*i.e.* , it is not necessary to store 14 seconds of values if the maximum delay required for this transition is 12 seconds) the older values start being overwritten. The type of data used in these arrays can be changed in the $typedef$ present in the generated $Functions.h$ file.

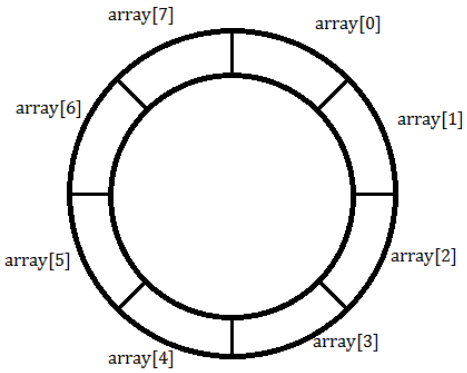

Figure 3.6: Circular Array

Chapter 4

Real System Application

In this chapter we are interested in applying this method in a real system in figure 4.1 which is located in Institut des sciences et techniques de l'ingénieur d'Angers (ISTIA) Angers, France.

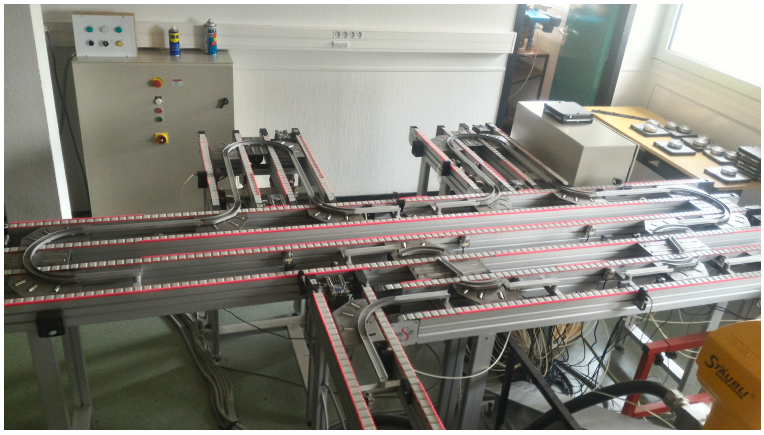

Figure 4.1: Real system

The system has 2 separated parts, a faster loop and a slower loop as showed in figure 4.1. The slower loop has 6 buttons that do not let the pallets pass while the faster loop has only 4. All the buttons has sensors just before them, as it can be seen in figure 4.2. The pallet's size is such that if they are waiting the button, the sensor will stay active. Each trajectory (between two buttons) has a defined maximum number of pallets. The travel times were measured 10 times, and the time used were the average of them.

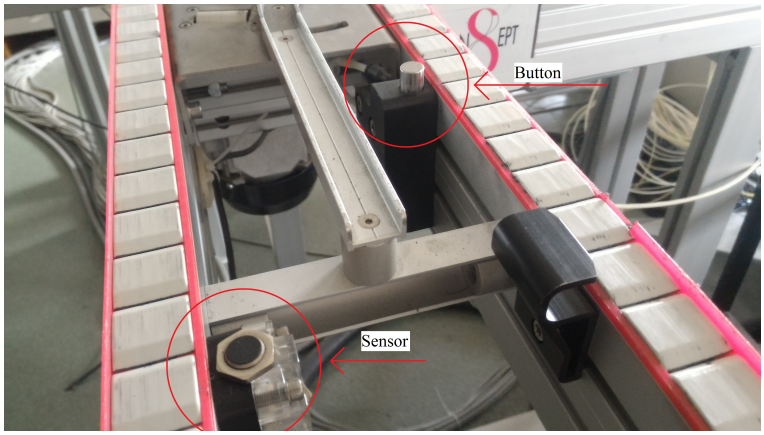

Figure 4.2: Button and Sensor

Each button was named B1 to B10 and the travel time as well as the maximum number of pallets are represented in Figure 4.3. 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.

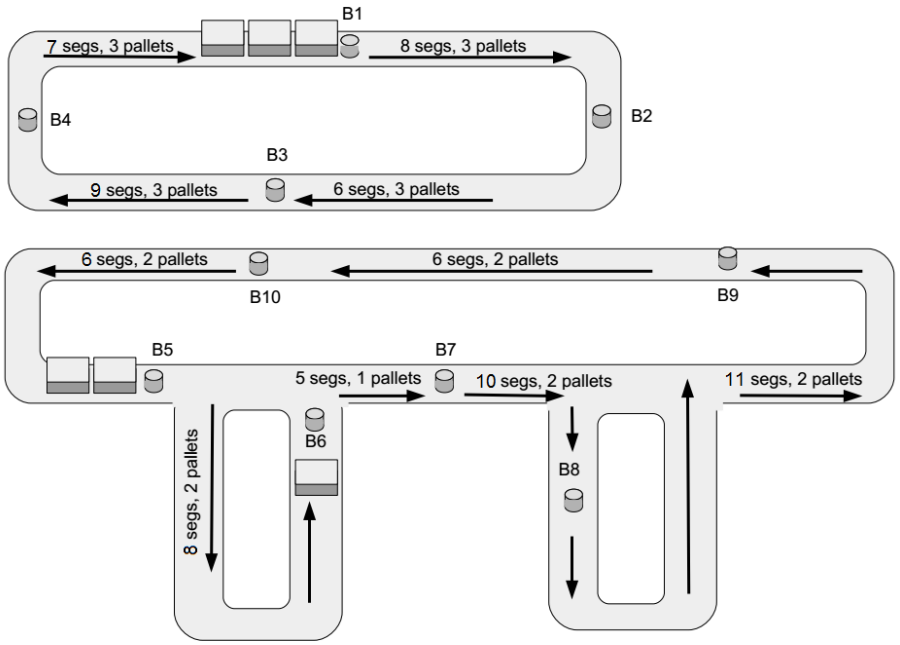

Figure 4.3: Buttons, travel times and pallet limit

The system is programmed to activate the buttons when there is one pallet waiting for it (the sensor is active), at least one space left for the path and at least one control token (that will be given by the supervisory system) 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 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 path between B4 and B1 and then apply the same reasoning to the other paths. 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 figure 4.4. Then we will need one place for each empty slot in the path, 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 traveling time. Using these conditions we have the second step of the model in figure 4.5. It is worth mentioning that the initial pallets are the tokens into (P1), (P2) and (P3) so that the token in (P1) can activate B1 instantaneously.

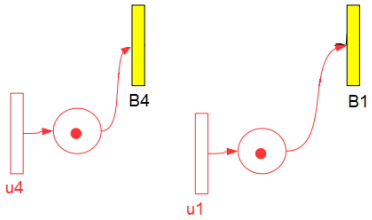

Figure 4.4: Modeling 1

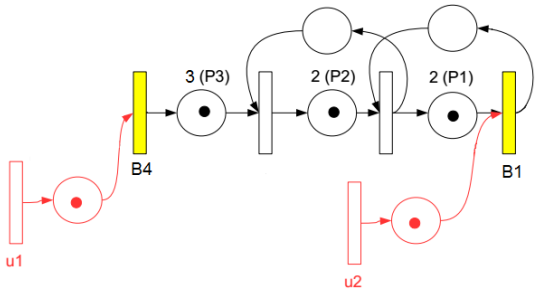

Figure 4.5: Modeling 2

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 traveling time in the model. The last condition that is missing is the total number of pallets for each path. Since for the path B4 to B1 only 3 pallets are allowed, and all the three are already in it, so we have the final model represented in figure 4.6.

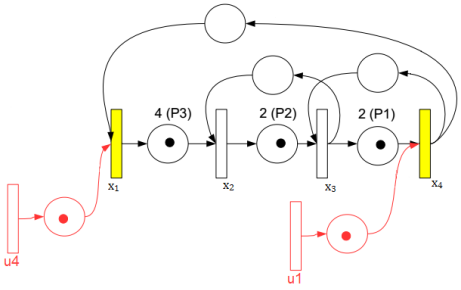

Figure 4.6: Final path Model

Lastly, the output y is made by the sensor, this means, the transition y is activated when the sensor just behind the button is activated. This is the reason why the output has to be put in the auxiliary state x_3 before the button transition, otherwise the output would be incremented when the button activates. If we replicate this reasoning to all paths combined we get the final model represented in figure 4.7. The image below as well as the model explanation were obtained in [11].

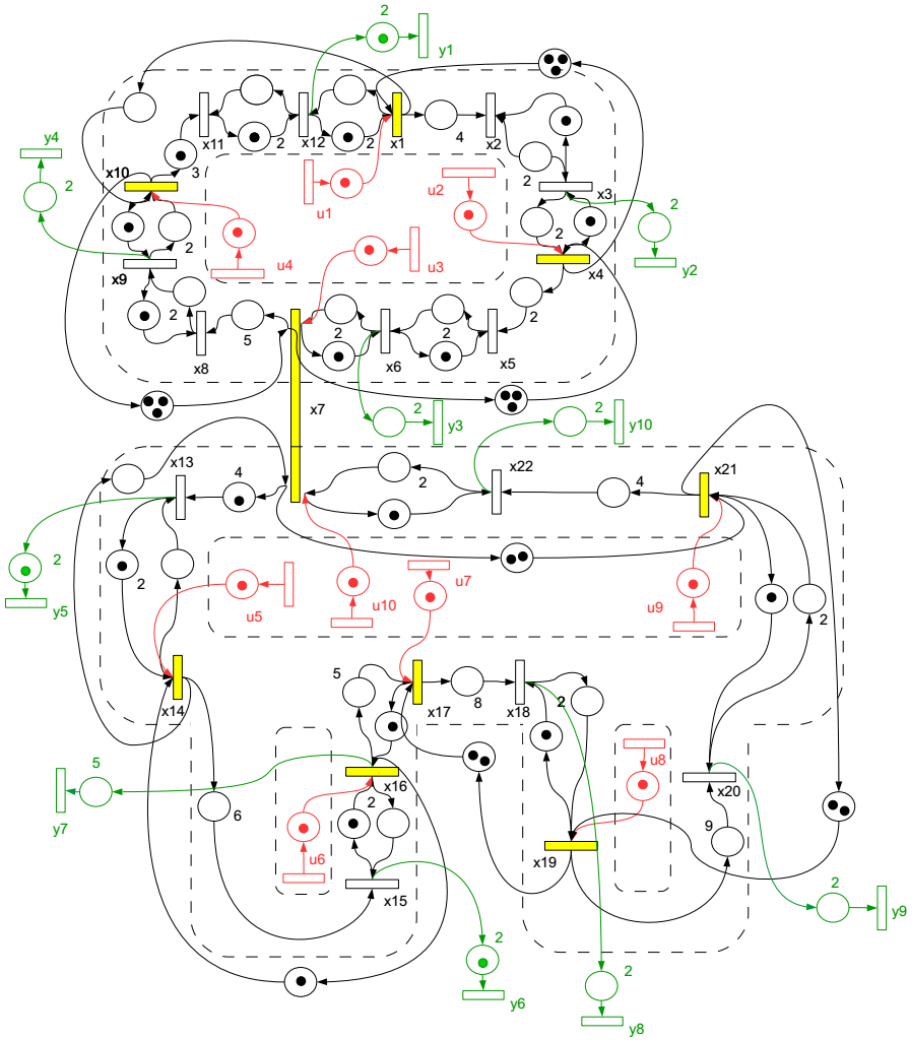

Figure 4.7: Complete Model

We can achieve the system's matrices as shown in appendix, where $\widetilde{A}$ will be devised into A_g, A_d and A_r to make the model implementable using the same procedure discussed in chapter 2. The next step is to use these 3 matrices to calculate P_{opt}, L_{opt} and M_{opt} thought equations 3.2-3.4. These calculus were made using the software described in chapter 3, using the data type as unsigned short int, a 7 thread input/output main program (3

threads for writing, 3 for reading and 1 thread to end the program) and an update control time of 500 ms. The process of calculus and implementation is shown in figure 4.8.

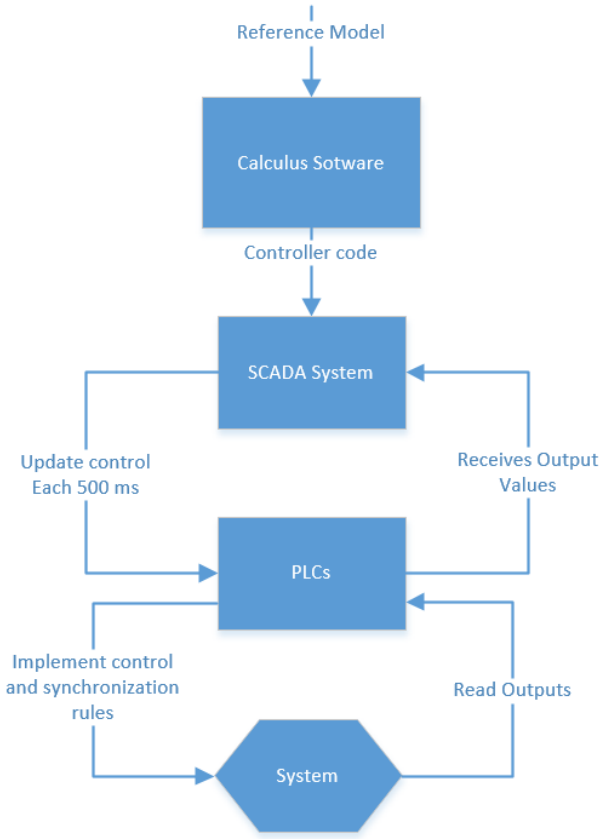

Figure 4.8: Implementation Process

In this way, the PLCs read the sensor's outputs and count the transitions y_1 to y_{10}, this information is then read by the SCADA System present in a computer, and the values for the observer and control tokens are updated. After updating those values, the computer waits for a certain time in a way that the inactive time plus the time taken to read/write the values and update all the values is equal to 500ms, basically $T_{update} + T_{idle} = 500ms$. With the control tokens counter updated, the PLCs start activating the buttons in order to control the system and minimize accumulation.

Chapter 5

Results

In this chapter 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 B3 and B10 and the loop containing B3 is faster, it is the only place that can have accumulation. G_{ref} is chosen such that it is equal to the system's transfer function, that means, we chose to maintain the original output, but delay as much as possible the buttons activations in order to reduce stock. Knowing that, the objective of the controller is to reduce the stock in front of B3 without changing the original output. The PLC was programmed to inform when the inner state x_7 was activated since all the outputs are defined by this state but with a time delay. In this matter, analyzing x_7's behavior we are able to analyze all outputs, meaning that if the objective is to maintain all outputs, the inner state x_7 without and with control must be the same.

Although, being exactly equal in this system is impossible since the traveling times are not deterministic, instead we expect the difference between x_7s to be floating around zero, sometimes positive and sometimes negative. In figure 5.1 it is possible to see the time evolution of the difference between x_7 before and after control. 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 gets bigger though 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 has perfectly deterministic.[1] For a visual explanation, the reader is invited to see the demonstration of the Not Controlled System and the Controlled System.

[1]Figures generated using MATLAB R2015b

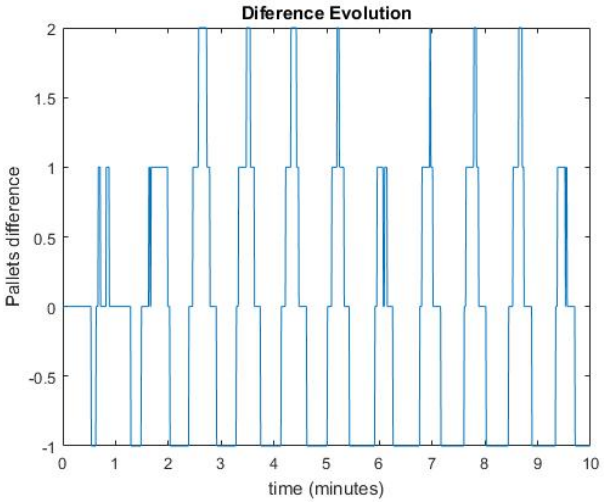

Figure 5.1: x_7's difference

The next topic will be the stocking difference. For this matter the PLCs were programmed to acquire 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 figure 5.2.

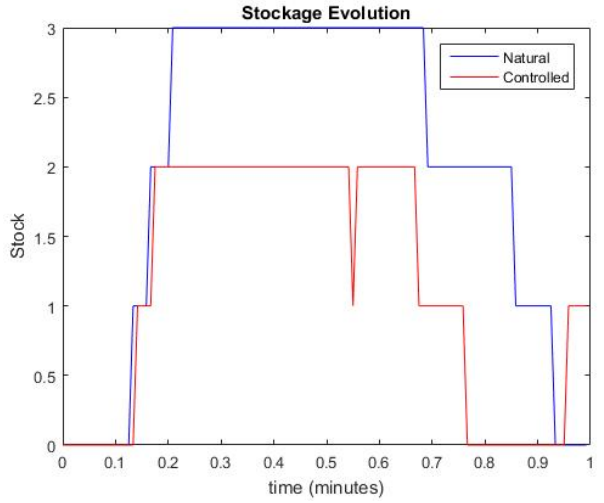

Figure 5.2: Stock Evolution

This pattern keeps repeating for the whole simulation. As it can be seen, the controlled system has at maximum 2 pallets in stock, while the not controlled (natural) system 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, in order to activate the sensor before B2 three times, 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. It is worth mentioning that this result is a direct consequence of our G_{ref} choice, another one can be chosen in order to have a different control and thus a different result. Although the $G_{ref} = H$ is hardcoded into the software, it is easy to change it to have a generic reference transfer matrix.

Next, We are going to analyze the effects of a malfunction in the system. A disturbance was put between buttons B7 and B8 in a way that the pallets could not pass though, this way button B3 and B10 would never activate. This test was made in both controlled and not controlled system. In figure 5.3 we can see the difference in stock between the controlled and not controlled systems. As expected the feedback controller still grants minimum stock even with a malfunction in the system. After, the disturbance was removed and the system began reacting as normal as presented in figure 5.2.

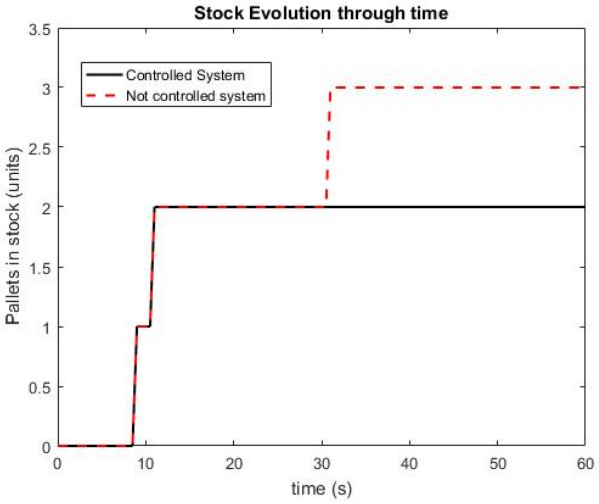

Figure 5.3: Stock Evolution in B3 with malfunctioning

Finally we made the same test, but this time putting the disturbance between B1 and B2, in order to see the accumulation in B10. Again, the test was made both in the controlled and not controlled systems. The results can be seen in figure 5.4. In this case, the controller made no difference in the accumulation time, this effect is due to the limitations of the system, this means, the path between B9 and B10 only accepts 2 pallets maximum. In other hand, when analyzed, the number of control tokens available for B9 was 0 in the controlled system, and infinity for the not controlled one, meaning that if the pallet restriction did non exist, the controlled system would have 2 pallets stock, and the not controlled one 3. The explanation for the required number of pallets is the same as for output y_2.

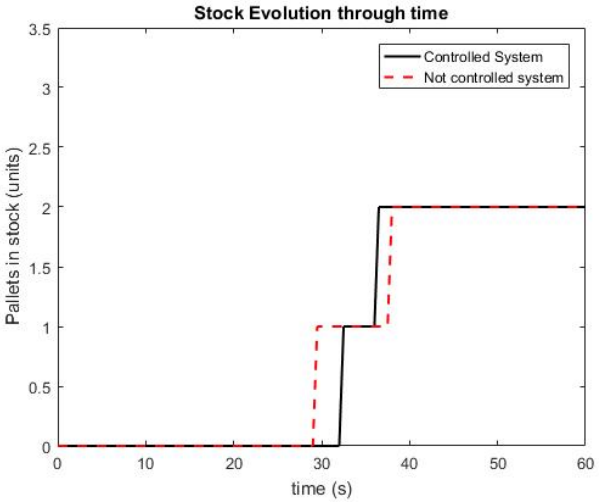

Figure 5.4: Stock Evolution in B10 with malfunctioning

Conclusion

After all these tests and simulations applying the chosen reference model $G_{ref} = H$, we can conclude the system had its pallet accumulation significantly reduced and the outputs have been maintained as expected. In this aspect, the software developed in the Institut des sciences et techniques de l'ingénieur d'Angers (ISTIA) made it easy to test and implement in any industrial line, with the only difficulty being modeling the system into TEG, and then translating this model to gamma-delta matrices.

An interesting progression to this study is to use the theory based on [12] to generate a reference model G_{ref} capable of limiting both the time in each inner state and the amount of pallets allowed for each place. In the system presented, it would be interesting to limit the amount of pallets in B3 and B10 to 1 and the hold time to 5 seconds, this would generate a controller with no stock in these places. This theory would only change the code that calculates the controller M, the observer L and the input filter P, the code to translate it into C can be used as normal.

Another interesting improvement is to make a better software, with a fully automatic code generation (the code made still requires the user to make the loop and the input/output with the controllers) as well as a graphic interface. This interface would let the user to insert the model, and do the gamma-delta modeling automatically, saving time and possible human errors.

Another problem is that the holding times must be divisible by the update control period, *i.e.* if the update time is 3 seconds and the holding time is 10 seconds, since the machine reads and stores the values each 3 seconds it is impossible to have 10 second delay, the program would pick the 12 second delay since the result is rounded up. It is worth mentioning that we were not able to think in a solution to this problem for every update rate, and the solution we used was to pick an update time that is a divisor of 1 second (in this case 500ms). The update time must be 100ms, 250ms, 500ms or 1s if all the holding times are integers.

Appendix

Model matrices for the real system

$$\widetilde{A} = \begin{bmatrix}
\varepsilon & \varepsilon & \varepsilon & \gamma^3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\delta^4 & \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \delta^2 & \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon & \gamma^3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon & \gamma^3 & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^5 & \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta^3 & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta^4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^6 & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta^2 & \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^5 & \varepsilon & \varepsilon & \gamma^2 & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^8 & \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \gamma^2 & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^9 & \varepsilon & \gamma & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^4 & \varepsilon
\end{bmatrix}$$

$$\widetilde{B} = \begin{bmatrix}
0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon
\end{bmatrix}$$

$$\widetilde{C} = \begin{bmatrix}
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \gamma^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\gamma\delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^5 & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2
\end{bmatrix}^T$$

Bibliography

[1] C. A. Maia, L. Hardouin, R. Santos Mendes, and B. Cottenceau. On the Model Reference Control for Max-Plus Linear Systems. In *Proceedings of the 44th*, pages 7799–7803, Seville, Spain, December 2005.

[2] M. Lhommeau, L. Hardouin, B. Cottenceau, and L. Jaulin. Interval Analysis and Dioid: Application to Robust Controller Design for Timed Event Graphs. *Automatica*, 40(11):1923–1930, November 2004.

[3] M. Lhommeau, L. Hardouin, and B. Cottenceau. Optimal Control for (max,+)-linear Systems in the Presence of Disturbances. *Positive Systems: Theory and Applications, POSTA, Springer LNCIS 294*, pages 47–54, 2003.

[4] B. Heidergott, G.J. Olsder, and J.W. van der Woude. *Max Plus at Work: Modeling and Analysis of Synchronized Systems : a Course on Max-Plus Algebra and Its Applications*. Number v. 13 in Max Plus at work: modeling and analysis of synchronized systems : a course on Max-Plus algebra and its applications. Princeton University Press, 2006.

[5] G. Cohen, S. Gaubert, and J.-P. Quadrat. Max-plus algebra and system theory : Where we are and where to go now. In *IFAC Conference on System Structure and Control*, Nantes, 1998.

[6] C. A. Maia, L. Hardouin, R. Santos Mendes, and B. Cottenceau. Optimal Closed-Loop Control for Timed Event Graphs in Dioid. *IEEE Transactions on Automatic Control*, 48(12):2284–2287, 2003.

[7] F. Baccelli, G. Cohen, G. J. Olsder, and J. P. Quadrat. *Synchronization and Linearity : An Algebra for Discrete Event Systems*. Wiley and Sons, 1992.

[8] Laurent Hardouin Rafael Santos-Mendes Renato Markele Ferreira Cândido, Mehdi Lhommeau. Minmaxgdjs : A web toolbox to handle periodic series in min-max[gamma,delta] semiring. In *IFAC World Congress*, 2017.

[9] D. G. Luenberger. An Introduction to observers. *IEEE Trans. on Automatic Control*, 16(6):596–602, 1971.

[10] L. Hardouin, Y. Shang, C. A. Maia, and B. Cottenceau. Observer-based controllers for max-plus linear systems. *IEEE Transactions on Automatic Control*, 62(5):2153–2165, May 2017. doi:10.1109/TAC.2016.2604562.

[11] V. Mariano Gonçalves. *Tropical Algorithms for Linear Algebra and Linear Event-Invariant Dynamical Systems*. Thesis, Universidade Federal de Minas Gerais, November 2014.

[12] T. Brunsch. *Modeling and Control of Complex Systems in a Dioid Framework*. Thesis, Université d'Angers, January 2014.

Abstract — This article aims to present the fundamentals of a Timed Event Graph (TEG - a petri net subclass) and apply it on industrial lines. Using a TEG, this article proposes an observer's feedback controller, based on classic theory, and a discussion of its benefits. Later, this calculus are exemplified using a simple system and the reader is able to follow the calculations (made by hand) to better understand the algebra. Finally, this controller is applied (using a software made by the author) into a real system, that simulates an industrial line. It aims to maintain a reference output while delaying the input as much as possible, in order to reduce overload in machines and lines, as well as other advantages.

Key words : Timed Event Graphs, Max-Plus, Control, State Observer, just-in-time.

ISTIA
62, avenue Notre Dame du Lac
49000 Angers