A non-linear set-membership approach for the control of Discrete Event Systems

M. Lhommeau [*] **L. Jaulin** [**] **L. Hardouin** [***]

[*] *LUNAM Université, Université d'Angers - Laboratoire d'ingénierie des systèmes automatisés (LISA) - ISTIA, 62 Avenue Notre Dame du Lac, 49000 ANGERS, France. (e-mail: mehdi.lhommeau@univ-angers.fr).*
[**] *OSM, Lab-STICC, ENSTA-Bretagne, 29806 Brest. (e-mail: luc.jaulin@ensta-bretagne.fr).*
[***] *LUNAM Université, Université d'Angers - Laboratoire d'ingénierie des systèmes automatisés (LISA) - ISTIA, 62 Avenue Notre Dame du Lac, 49000 ANGERS, France. (e-mail: laurent.hardouin@univ-angers.fr).*

Abstract: A variety of problems in non-linear time-evolution systems such as manufacturing plants, operations research, computer networks, etc., can be modelled as min-max-plus systems in which operations of min, max and addition appear simultaneously. It is well know that systems with only maximum (or minimum) constraints can be modelled as max-plus system and handled by max-plus algebra which changes the original non-linear system into linear system in this framework. Several authors have developed methods, in max-plus algebra, to compute control for max-plus systems. In general, these methods use the residuation theory to design a just-in-time control such that the output of the controlled system is, on the one hand, less than the desired reference signal but as close as possible to the given reference and, on the other hand, the control is delayed as much as possible. In this paper, we consider min-max-plus systems which are extensions of max-plus systems and non-linear even in the max-plus algebra view. We proposes a new approach to solve the just-in-time control problem for a non-linear min-max-plus system. This problem is cast into the more general framework of constraint satisfaction problems. This makes it possible to propose a new algorithm that contracts the feasible domains for each uncertain variable optimally (i.e., no smaller domain could be obtained) and efficiently.

Keywords: Max-plus Algebra, Timed Event Graphs, Min-Max-Plus, Interval Arithmetic, Constraint Satisfaction Problem

1. INTRODUCTION

This paper proposes a set-membership approach to deal with the control of Discrete Event Systems (DES). Many engineering systems such as manufacturing (Ayhan and Wortman (1999)), transport and communication networks (Boudec and Thiran (2002)) and others, can be modelled by using the Discrete Event Systems and in particular as max-plus systems. Systems modelled as max-plus systems can be handled by max-plus algebra methods which have the great advantage of changing the original non-linear systems into linear systems in the view of max-plus algebra framework. Among them it can be noticed results about performance analysis and controller synthesis. This kind of systems can be represented graphically by using Timed Event Graph (TEG), which is a particular class of timed Petri net in which all places have single upstream and single downstream transitions (Murata (1989)). TEG control problems are usually stated in a Just-in-time context. The goal is to achieve some performance while minimizing internal stocks (Cohen et al. (1998); Menguy et al. (2000); Maia et al. (2003)). In general, max-plus system model can only handle problems with maximum (or minimum) constraints which are linear in the max-plus algebra view. However, there exist a lot of problems with both minimum and maximum constraints in the real world (see Baccelli et al. (1992), Gunawardena (1994), Zhu et al. (2009)). These non-linear time-evolution systems can be described by min-max-plus systems which include max-plus systems as a special case and are non-linear even in the max-plus algebra view. There has been much research on min-max-plus systems in recent years. Some significant results have been obtained for autonomous systems, such as the existence and the calculation of a fixed point and a cycle time (see Gunawardena and Keane (1995), Wen-De Chen (2010) ; etc.). The control problems of min-max-plus systems have been studied lately. De Schutter and van den Boom investigated the model predictive control for min-max-plus systems (see De Schutter and van den Boom (2000); Necoara et al. (2007)); Chen and Tao investigated the observability, reachability and cycle time assignment Chen and Tao (2001). In this paper, we study extensions of just-in-time control of max-plus systems to min-max-plus systems. Such systems are non-linear in both the max-plus and the min-plus algebra, then it is not possible to use conventional tools such as the theory of residuation. Then,

we propose a new approach based on interval arithmetic and constraint propagation methods for analysing the control of min-max-plus systems. This paper is organized as follows. In section 2, notations and background concerning min-max-plus systems is given. Section 3 introduces Timed Event Graphs description and the just-in-time control problem. Section 4 will recall the basic notions of interval analysis and constraint propagation needed to solve the control problems of DES. Then we illustrate the constraint propagation using an example that can be described by linear equations in max-plus algebra. In Section 5, we apply the methods of constraint propagation on a just-in-time control problem for min-max-plus system.

2. PRELIMINARIES

2.1 General min-max-plus systems

Let us begin with some notations which are used through this paper. Let $\mathbb{R}$ be a set of all real numbers and $\mathbb{R}^n$ be an n-dimensional column vector set over $\mathbb{R}$. Vectors in $\mathbb{R}^n$ are denoted by a bold lower-case letter, $e.g.$, $\mathbf{x}$ and x_j denote the j-th component of $\mathbf{x}$. The notation $\mathbf{x} \le \mathbf{y}$ denotes the usual partial order on $\mathbf{R}^n$, $i.e.$,

$$\mathbf{x} \le \mathbf{y} \iff x_j \le y_j, \text{ for } 1 \le j \le n.$$

The notations $a \wedge b$ and $a \vee b$ stand for minimum and maximum of real numbers a and b, respectively, $i.e.$,

$$a \wedge b = \min(a, b), \quad a \vee b = \max(a, b).$$

Note that $+$ distributes over both $\wedge$ and $\vee$, then

$$(a \wedge b) + c = (a+c) \wedge (b+c), \quad (a \vee b) + c = (a+c) \vee (b+c).$$

In the sequel, it is assumed that $+$ always has higher precedence than either $\wedge$ or $\vee$. The equalities above can hence be rewritten as

$$(a \wedge b) + c = a + c \wedge b + c, \quad (a \vee b) + c = a + c \vee b + c.$$

The operations $\wedge$ and $\vee$ are also used for the corresponding operations on vectors :

$$(\mathbf{x} \wedge \mathbf{y})_j = x_j \wedge y_j \text{ and } (\mathbf{x} \vee \mathbf{y})_j = x_j \vee y_j.$$

A min-max-plus function of type $(n, 1)$ is any function $f : \mathbb{R}^n \to \mathbb{R}^1$, which can be written as a term in the following grammar :

$$f = x_1, \dots x_n \mid f + a \mid f \wedge f \mid f \vee f, \quad (1)$$

where $x_1, \dots, x_n$ are variables, and $a \in \mathbb{R}$ is referred to as a parameter. The vertical bars separate the different ways in which terms can recursively be constructed. The simplest term is one of the n variables, x_j, thought of as the j-th component function. Given any term, a new one may be constructed by adding $a \in \mathbb{R}$; given two terms, a new one may be constructed by taking the minimum or the maximum. Only these rules may be used to build terms. For example, $(x_1 + 5 \vee x_2 + 8) \wedge x_3$ is a min-max-plus function of type $(3, 1)$, neither $x_1 \vee 4$ nor $(x_1 + x_2) \wedge (x_3 + 3)$ can be generated by grammar (1).

A min-max-plus function of type (n, m) is any function $F : \mathbb{R}^n \to \mathbb{R}^m$ such that each component is a min-max-plus expression of n variables $x_1, \dots, x_n$, and denoted by $F_i(\mathbf{x})$. The set of min-max-plus functions of type (n, m) is denoted by $MMP(n, m)$. A min-max-plus expression which uses only $\vee$ and $+$ is said to be max-plus only. A min-max-plus function of type (n, m) is said to be max-plus only if its components are all max-plus only. Now,

recall some basic properties of min-max-plus functions. Let $F(\mathbf{x})$ be a min-max-plus function of type (n, m). First $F(\mathbf{x})$ is continuous in $\mathbf{x}$. Second, $F(\mathbf{x})$ is monotone : $\mathbf{x} \le \mathbf{y} \Rightarrow F(\mathbf{x}) \le F(\mathbf{y})$. Third, $F(\mathbf{x})$ is homogeneous, in the sense that, for any $h \in \mathbb{R}, F(\mathbf{x} + h) = F(\mathbf{x}) + h$.

A min-max-plus system is a system described by

$$\begin{cases} \mathbf{x}(k+1) = F(\mathbf{x}(k)) \vee G(\mathbf{u}(k)) \\ \mathbf{y}(k) \quad = C(\mathbf{x}(k)), \quad k = 0, 1, \dots \end{cases} \quad (2)$$

where $\mathbf{x}(0) = \zeta \in \mathbb{R}^n$, $F(\mathbf{x}) \in MMP(n, n)$, $G(\mathbf{u}) \in MMP(q, n)$ and $C(\mathbf{x}) \in MMP(n, p)$ are system, input and output functions respectively, and $\mathbf{x}(k) = [x_1(k), \dots, x_n(k)]^t \in \mathbb{R}^n$, $\mathbf{u}(k) = [u_1(k), \dots, u_q(k)]^t \in \mathbb{R}^q$ and $\mathbf{y} = [y_1(k), \dots, y_p(k)]^t \in \mathbb{R}^p$ are state, input and output vectors, respectively.

A max-plus (dually, min-plus) system is a min-max-plus system where all F_i components are max-plus only (respectively min-plus only) expressions.

Rather than writing a max-plus system as (2) one often writes

$$\begin{aligned} \mathbf{x}(k+1) &= A\mathbf{x}(k) \vee B\mathbf{u}(k) \\ \mathbf{y}(k) &= C\mathbf{x}(k) \end{aligned}$$

which is short-hand notation for

$$x_i(k+1) = \max(a_{i1} + x_1(k), \dots, a_{in} + x_n(k),$$
$$b_1 + u_1(k), \dots b_n + u_n(k)),$$
$$y_i(k) = \max(c_{i1} + x_1(k), \dots, c_{in} + x_n(k))$$
$$i = 1, 2, \dots, n. \quad (3)$$

3. CONTROL PROBLEMS

This section deals with Petri Nets (PN) and Timed Event Graphs (TEG) that provide an intuitive way of modelling discrete-event systems. In this paper we address the problem of the just-in-time control of discrete-event systems. This problem has been chosen for the following reasons (i) it illustrate, simply, how interval propagation techniques can be used for the control of a discrete event system. Moreover, when we consider a linear system in max-plus algebra, we obtain the same results as the residuation theory (ii) the constraint propagation techniques are more general than the residuation theory and can solve larger problems (for example, min-max-plus systems which are non-linear in (max-plus,min-plus algebra).

3.1 Timed Event Graphs (TEG)

A *Petri net* is a directed bipartite graph $(\mathcal{P}, \mathcal{T}, \mathcal{E}, \mathcal{F})$ where the set of nodes is divided into two categories : place in $\mathcal{P}$ and transitions in $\mathcal{T}$. The directed arcs lie either in $\mathcal{E} \subset \mathcal{P} \times \mathcal{T}$ or in $\mathcal{F} \subset \mathcal{T} \times \mathcal{P}$. *Tokens* circulate in such a Petri net. The number of tokens in each place at time t, constitute its *marking* at time t. A transition is *enabled* if each upstream place contains at least one token. The *firing* of an enabled transition removes one token to each upstream place and adds one token to each downstream place. *Event graphs* constitute a subclass of Petri net $i.e.$, those places have one and only one upstream and downstream transition. In a *Timed Event Graph* (TEG), a delay is associated to each place. This delay is the time
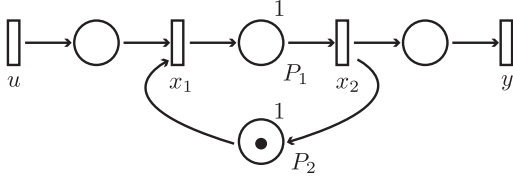
Fig. 1. A timed event graph

that a token must stay in the place before contributing the enabling of the downstream transition. On these topic the reader is referred to Baccelli et al. (1992). Timed event graphs are particularly suitable to model processes which require synchronizations and delays. For example, the timed event graph of Figure 1 represents a production line. The delay of place P_1 represents the basic production time and the delay of place P_2 (one unit) is the time necessary to initialize the production line.

Two formalisms are commonly used to represent the behaviour of a TEG. On the one hand, for each transition x_i, we define the function $x_i(k)$ called *dater*, which represents the date at which the transition x_i has been fired for the kth time. On the other hand, one is interested in the number of times transitions x_i has been fired at time t ; this function $x_i(t)$ associated to transition x_i is called *counter*. Daters and counters respectively lead to system in the min-plus algebra (or $\overline{\mathbb{Z}}_{max}$) and in the min-plus algebra (or $\overline{\mathbb{Z}}_{min}$).

In the example of Figure 1, the vector $\mathbf{x}(k) = (x_1(k), x_2(k))'$ is the state of the system. The daters of the transitions without upstream place form the vector $u(k)$ which is the input of the system, and the output vector $y(k)$ is formed by the daters of the transitions without downstream transition. Then the above graph is described by the following system

$$\begin{cases} x_1(k) = \max(u(k), x_2(k-1)+1) \\ x_2(k) = x_1(k)+1 \\ y(k) = x_2(k) \end{cases} \quad (4)$$

That can be also written as a linear system in the max-plus algebra, that is

$$\begin{cases} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} = \begin{pmatrix} -\infty & 1 \\ -\infty & 2 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} \vee \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(k) \\ y(k) = (-\infty \; e) \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} \end{cases}$$

In the sequel, we will first address the following control problem for max-plus linear systems.

Problem 1 : Let q be a finite integer. Given system (4) and a desired output transition firing date $z = \{z(k)\}_{(k \in \underline{q})}$, we search for the *latest* input transition firing dates $u = \{u(k)\}_{(k \in \underline{q})}$ such that output transition firing dates $y = \{y(k)\}_{(k \in \underline{q})}$ of (4) satisfies $y(k) \preceq z(k)$, for all k in $\underline{q}$?

This tracking problem formally consists in finding the greatest solution to inequality $H(u) \preceq z$ where H is a max-plus linear operator and $H(u) = \{[H(u)]\}_{(k \in \underline{q})}$ is the output associated with input u. In a production context, such a control minimizes the work in process (find the greatest control) while satisfying the customer demand (the output is less than or equal to the reference output).

This problem is widely known and an optimal control exists when the model is exact (Baccelli et al. (1992)), its synthesis is based on an open-loop control structure and principally uses the residuation theory (Blyth and Janowitz (1972)).

Remark : For our simulation the desired output transition firing date $z(k)$ is given by

k	0	1	2	3	4
$z(k)$	$[-\infty, 3]$	$[-\infty, 4]$	$[-\infty, 7]$	$[-\infty, 8]$	$[-\infty, 15]$

We suppose that the components of $x_1(0)$, $x_1(1)$, $x_1(2)$, $x_1(3)$, $x_1(4)$, $x_2(0)$, $x_2(1)$, $x_2(2)$, $x_2(3)$, $x_2(4)$, $u(0)$, $u(1)$, $u(2)$, $u(3)$, $u(4)$ are not measured, and thus associated prior intervals are $[-\infty, \infty]$.

Interval propagation is a set of numerical methods that make possible to contract the interval domains for the variables, without losing any feasible value. After a brief presentation of interval propagation, we will illustrate the efficiency of the approach on our simple control problem.

4. INTERVAL PROPAGATION

4.1 Interval arithmetic

An interval is a closed and connected subset of $\mathbb{R}$ defined as $[a,b] = \{x | a \le x \le b\}$. Endpoints of an interval $[x]$ are denoted by x^- and x^+. Thus, $[x] = [x^-, x^+]$. The interval $[x, x]$ is a *degenerate interval* which we do not distinguish from the number x. Consider two intervals $[x]$ and $[y]$ and an operator $\diamond \in \{+, -, \times, /\}$, we define $[x] \diamond [y]$ as the smallest interval which contains all feasible values for $x \diamond y$, if $x \in [x]$ and $y \in [y]$ (see Moore (1979)). For instance

$$\begin{aligned} [-1,3] + [2,5] &= [1,8] \\ [-1,3] \times [2,5] &= [-5,15] \\ [-1,3] / [2,5] &= \left[-\frac{1}{2}, \frac{3}{2}\right] \end{aligned}$$

If f is an elementary function such as $\sin, \cos, \min, \max, \ldots$ we define $f([x])$ as the smallest interval which contains all feasible values for $f(x)$.

Example 1. The max of two intervals (the definition can be generalized for n intervals) is defined as $\max(x,y) = [\max(x^-, y^-), \max(x^+, y^+)]$ and the min as $\min(x,y) = [\min(x^-, y^-), \min(x^+, y^+)]$.

An interval vector $[\mathbf{x}]$ is a vector whose components are intervals

$$[\mathbf{x}] = [x_1^-, x_1^+] \times \ldots \times [x_n^-, x_n^+] = [x_1] \times [x_n].$$

4.2 Contraction and propagation

Consider a constraint $\mathcal{C}$ (*i.e.*, an equation or an inequality), some variables $x_1, x_2, \ldots$ involved in $\mathcal{C}$ and prior interval domains $[x_i]$ that contain all feasible values for the x_i's. Interval arithmetic makes possible to contract the domains $[x_i]$ without removing any feasible values for the x_i's.

Example 2. Consider the equation

$$(C_1) : x_3 = x_1 + x_2$$

where the domains for x_1, x_2 and x_3 are given by $[x_1] = [-\infty, 5], [x_2] = [-\infty, 4]$ and $[x_3] = [6, \infty]$. These domains can be contracted with interval propagation :

$$x_3 = x_1 + x_2 \Rightarrow [6, \infty] \cap ([-\infty, 5] + [\infty, 4])$$
$$= [6, \infty] \cap [-\infty, 9] = [6, 9].$$
$$x_1 = x_3 - x_2 \Rightarrow [-\infty, 5] \cap ([6, \infty] - [-\infty, 4])$$
$$= [-\infty, 5] \cap [2, \infty] = [2, 5].$$
$$x_2 = x_3 - x_1 \Rightarrow [-\infty, 4] \cap ([6, \infty] - [-\infty, 5])$$
$$= [-\infty, 4] \cap [1, \infty] = [1, 4].$$

Then, we have the following contracted domains : $[x_1'] = [2, 5], [x_2'] = [1, 4]$ and $[x_3'] = [6, 9]$.

This contraction procedure can be performed with much more complex constraints. A contraction operator is called a *contractor* (see Chabert and Jaulin (2009)). The contraction illustrated above can be described by the following procedure, where CPLUS stands for Contractor of the constraint PLUS.

Algorithm CPLUS (inout : $[z], [x], [y]$)
1    $[z] = [z] \cap ([x] + [y])$
2    $[x] = [x] \cap ([z] - [y])$
3    $[y] = [y] \cap ([z] - [x])$

When more than one constraint are involved (see Example 3), the contractions are performed sequentially several times, until no more significant contractions can be observed. It can be shown that the interval vector to which the method converges does not depend on the order to which the contractors are applied (Montanari (1974)), but the computing time is highly sensitive to this order. There is no optimal order in general, but in practice, one of the most efficient is called *forward-backward propagation*. It consists in writing the whole set of equations under the form $\mathbf{f}(\mathbf{x}) = \mathbf{y}$ where $\mathbf{x}$ and $\mathbf{y}$ correspond to quantities can be measured (*i.e.*, some prior interval domains are given for them). Then, using interval arithmetic, the intervals are propagated from $\mathbf{x}$ to $\mathbf{y}$ in a first step (*forward propagation*) and, in a second step, the intervals are propagated from $\mathbf{y}$ to $\mathbf{x}$ (*backward propagation*).

Example 3. To illustrate the propagation process with several constraints, consider the three following equations

$$\begin{cases} (C_1): & y = x^2 \\ (C_2): & xy = 1 \\ (C_3): & y = -2x + 1. \end{cases}$$

Thanks to interval analysis, we want to prove that this system has no solution. To each of the variables, we assign the domain $[-\infty, \infty]$. Then we contract the domains with respect to the constraints in the following order : C_1, C_2, C_3, C_1, C_2 and we get empty intervals for x and y. A geometric interpretation of the propagation is given on Figure 2. The resulting interval computation is as follows.

$$(C_1) \Rightarrow y \in [-\infty, \infty]^2 = [0, \infty]$$
$$(C_2) \Rightarrow x \in 1/[0, \infty] = [0, \infty]$$
$$(C_3) \Rightarrow y \in [0, \infty] \cap ((-2) \times [0, \infty] + 1) = [0, 1]$$
$$x \in [0, \infty] \cap (-[0, 1]/2 + 1/2) = [0, 1/2]$$
$$(C_1) \Rightarrow y \in [0, 1] \cap [0, 1/2]^2 = [0, 1/4]$$
$$(C_2) \Rightarrow x \in [0, 1/2] \cap 1/[0, 1/4] = \emptyset$$

The interval propagation method converges to an interval vector which encloses all solutions (if there exists any), but the interval vector is not necessarily the smallest one. The interval vector is said to be *locally consistent* because it is consistent with all constraints taken independently. The smallest interval vector which encloses all solutions is said to be *globally consistent*. The problem of computing this
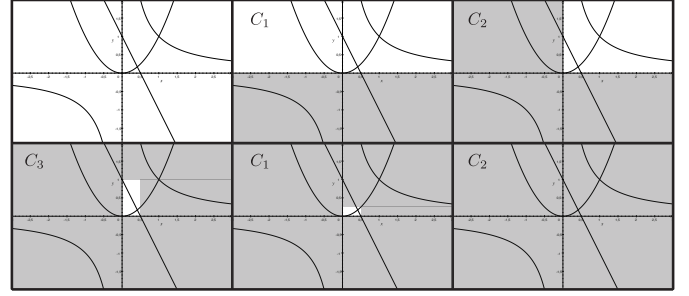

Fig. 2. Illustration of the propagation procedure

smallest box is NP-hard and can be solved using bisection methods only for problems involving few variables. In this paper, since we want to solve large dimensional problems, local consistency methods will be used.

4.3 Forward-Backward propagation for max-plus systems

To illustrate the principle, consider the equations (4). If $\mathbf{x} = (x(0), u(0), u(1), u(2), u(3), u(4))$ and $\mathbf{y} = (y(0), y(1), y(2), y(3), y(4), x(1), x(2), x(3), x(4))$, then (4) can be rewritten under the form $\mathbf{f}(\mathbf{x}) = \mathbf{y}$. The forward contraction can be described by the following algorithm

FORWARD CONTRACTION
1   **for** k=0 **to** 4
2     $[w](k) = [x_2](k-1) + 1$
3     $[x_1](k) = \max([u](k), [w](k))$
4     $[x_2](k) = [x_1](k) + 1$
5     $[y](k) = [x_2](k)$
6   **end**

where for $k = 0$, $[x_2](k-1) = [-\infty, +\infty]$. The backward propagation is described by

BACKWARD CONTRACTION
1   **for** k=4 **to** 0
2     $[y](k) = [y](k) \cap ([z](k) - [e](k))$
3     $[z](k) = [z](k) \cap ([y](k) + [e](k))$
4     $[x_2](k) = [x_2](k) \cap [y](k)$
5     $[x_1](k) = [x_1](k) \cap ([x_2](k) - 1)$
6   CMAX$([x_1](k), [u](k), [w](k))$
7     $[x_2](k-1) = [x_2](k-1) \cap ([w](k) - 1)$
8   **end**

where the inequality constraint $y(k) \preceq z(k)$ is transformed in equality by using a slake variable $[e](k) = [0, \infty], \forall k$, *i.e.*, $[y](k) + [e](k) = [z](k)$. The contractor for the constraint $[z] = \max([x], [y])$ is evaluated by the following algorithm.

Algorithm CMAX (inout : $[z], [x], [y]$)
1   **if** $([z] \cap [x] = \emptyset)$ **then** $[y] = [y] \cap [z]$
2   **else if** $([z] \cap [y] = \emptyset)$ **then** $[x] = [x] \cap [z]$
3   $[x] = [x] \cap [-\infty, z^+]$
4   $[y] = [y] \cap [-\infty, z^+]$

4.4 Results for the just-in-time control of TEG

Consider again the simple control problem (Problem 1) example of Section 3. If we apply an elementary interval propagation, we get the contracted intervals given in the following table.

k	0	1	2	3	4
Before propagation					
$[z](k)$	$[-\infty,3]$	$[-\infty,4]$	$[-\infty,7]$	$[-\infty,8]$	$[-\infty,15]$
$[y](k)$	$[-\infty,\infty]$	$[-\infty,\infty]$	$[-\infty,\infty]$	$[-\infty,\infty]$	$[-\infty,\infty]$
$[x_1](k)$	$[-\infty,\infty]$	$[-\infty,\infty]$	$[-\infty,\infty]$	$[-\infty,\infty]$	$[-\infty,\infty]$
$[x_2](k)$	$[-\infty,\infty]$	$[-\infty,\infty]$	$[-\infty,\infty]$	$[-\infty,\infty]$	$[-\infty,\infty]$
$[u](k)$	$[-\infty,\infty]$	$[-\infty,\infty]$	$[-\infty,\infty]$	$[-\infty,\infty]$	$[-\infty,\infty]$
After propagation					
$[z](k)$	$[-\infty,2]$	$[-\infty,4]$	$[-\infty,6]$	$[-\infty,8]$	$[-\infty,15]$
$[y](k)$	$[-\infty,2]$	$[-\infty,4]$	$[-\infty,6]$	$[-\infty,8]$	$[-\infty,15]$
$[x_1](k)$	$[-\infty,1]$	$[-\infty,3]$	$[-\infty,5]$	$[-\infty,7]$	$[-\infty,14]$
$[x_2](k)$	$[-\infty,2]$	$[-\infty,4]$	$[-\infty,6]$	$[-\infty,8]$	$[-\infty,15]$
$[u](k)$	$[-\infty,1]$	$[-\infty,3]$	$[-\infty,5]$	$[-\infty,7]$	$[-\infty,14]$

The computing time to get the contracted intervals is always less than 0.1 sec on a standard laptop. Finally, we can choose any control $u_0 \in [u](0)$, since any instantiation of u_0 does not change the domain for $u_1 \in [u](1)$. Note that, it is possible to choose control sequences that are not monotonic. Of course, such decreasing sequences have no physical meaning for a TEG and should not be chosen. Then, for our example (Problem 1), the optimal input is given by

$$(u(0)\ldots u(4)) = (1,3,5,7,14).$$

In a production context, such a control minimizes the work in process while satisfying the customer demand, that is the output is less than or equal to the reference output $y(k) \preceq z(k)$ for all k.

Remark 1. It is not difficult to show that, the step of Backward propagation gives the same results that we obtain in max-plus algebra with the residuation theory (Baccelli et al. (1992)). This is due to the fact that the inverse image of a cone of $\mathbb{R}^n$ of the form $[\mathbf{x}] = [-\infty, x_1^+] \times \ldots \times [-\infty, x_n^+]$ by a max-plus or min-plus function is still a cone. In other words, the max-plus or min-plus function are cone conservative (Jaulin et al. (2004)). Then, if all domains are cones, the consistency domains can be computed exactly provided that only corner-conservative functions are involved in the set propagation. Moreover, in this case the interval propagation method converges in one step.

In this section, we have shown that constraints propagation and interval computation, with an another point of view, can solve control problems for TEG with guaranteed solutions.

5. CONSTRAINT PROPAGATION AND MIN-MAX-PLUS SYSTEM CONTROL

The aim of this section is to introduce the application of constraints propagation techniques presented in Section 4 to solve a just-in-time control problem in the context of min-max-plus systems.

Problem 2: Consider a manufacturing system which consists of two machines, denoted by M_1 and M_2, one input, denoted by u_1 by which users can exert control on the system, and two outputs, denoted by y_1 and y_2, from which users can examine the outcome of the system. The two machines work recursively over the time : (1) the instruction $u_1(k)$ arrives; (2) A product carrier from M_1 or from M_2 finished its $(k-1)$th processing and arrives at M_1. M_2 starts a processing when a product carrier from M_1 finished its $(k-1)$th processing and arrives at M_2. y_1

is obtained when a product carrier from M_2 finished its kth round processing and arrives at y_1.
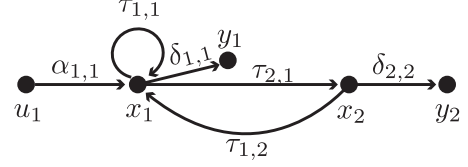

Fig. 3. A production line

The above system can be modelled as the min-max-plus system :

$$\begin{cases} \mathbf{x}(k+1) & = F(\mathbf{x}(k)) \vee G(\mathbf{u}(k)) \\ \mathbf{y}(k) & = C(\mathbf{x}(k)), \quad k = 0,1,\ldots, \\ \mathbf{x}(0) & = \xi \in \mathbb{R}^n \end{cases} \quad (5)$$

where $F(\mathbf{x})$, $G(\mathbf{u})$ and $C(\mathbf{x})$ are given by

$$\begin{aligned} F_1(\mathbf{x}) &= \tau_{1,1} + x_1 \wedge \tau_{1,2} + x_2, \\ F_2(\mathbf{x}) &= \tau_{2,1} + x_1, \end{aligned}$$

$$G_1(\mathbf{u}) = \alpha_{1,1} + u_1,$$

and

$$\begin{aligned} C_1(\mathbf{x}) &= \delta_{1,1} + x_1, \\ C_2(\mathbf{x}) &= \delta_{2,2} + x_2, \end{aligned}$$

respectively, where x_i denotes the processing finish time of the machine M_i in the kth round, $\tau_{i,j}$ denote the fixed time it takes by a product carrier which transfers from the M_j in the $(k-1)$th round and to M_i in the kth round, $\alpha_{i,j}$ denote the fixed time it takes by an input instruction which transfers from the u_j in the kth round and to M_i in the kth round, and $\delta_{i,j}$ denote the fixed time it takes by a product carrier which transfers from the M_j in the kth round and to y_i in the kth round.

Suppose $\tau_{1,1} = 3, \tau_{1,2} = 2, \tau_{2,1} = 2, \alpha_{1,1} = 2, \delta_{1,1} = 2, \delta_{2,2} = 3$, we take the desired outputs trajectories given below.

k	0	1	2	3	4
$z_1(k)$	$[-\infty,-\infty]$	$[-\infty,4]$	$[-\infty,9]$	$[-\infty,11]$	$[-\infty,13]$
$z_2(k)$	$[-\infty,-\infty]$	$[-\infty,-\infty]$	$[-\infty,7]$	$[-\infty,12]$	$[-\infty,14]$

All other domains are taken as $[-\infty,\infty]$, application of Algorithm JITCMMPS (for Just-In-Time Control for Min-Max-Plus System, Table A.1) yields the following contractions for the control input $u(k)$.

k	0	1	2	3	4
$u(k)$	$[-\infty,0]$	$[-\infty,5]$	$[-\infty,7]$	$[-\infty,9]$	$[-\infty,+\infty]$

Like the Problem 1, we can choose any control $u_0 \in [u](0), \ldots, u_k \in [u](k)$. Then, the greatest input control such that $(y_1(k) \preceq z_1(k)$ and $(y_2(k) \preceq z_2(k)$ for all $k = [0, \overline{k}]$ is given by

$$(u_0, \ldots, u_4) = (0,5,7,9,\infty).$$

In Algorithm JITCMMPS (see Table A.1 in Appendix A), we need a contractor for constraint $[z] = \min([x],[y])$. This contractor is given by Table A.2 in Appendix A.

The experiments were carried out with Matlab 7.12.0 (R2011a) with the support of the packages INTLAB 6 (see

Rump (1995)). INTLAB 6 provides interval arithmetic and useful interval functions. Thus, all of our computations were reliable and the results are verified.

6. CONCLUSION

This paper proposes a new and general algorithm to compute a control sequence, in a just-in-time context, of a min-max-plus system. The approach is based on constraint propagation and interval arithmetic. The methodology has been illustrated by two examples. The problem 1 can be solved by using the residuation theory, but residuation only applies to Timed Event Graphs, which represent a tiny class of DES. In contrast, to our knowledge, Problem 2, could not to be solved rigorously by other existing methods.

Appendix A. ALGORITHM JITCMMPS

Algorithm JITCMMPS
$(\textbf{in} : [z_1](0), \dots [z_1](\overline{k})], [z_2](0), \dots [z_2](\overline{k})]; \textbf{out} : [u_1](0), \dots, [u_1](\overline{k}))$
1   **do**
2   **for** $k=0$ **to** $\overline{k}$
3      $[w_1](k) = [x_1](k) + \tau_{1,1}$
4      $[w_2](k) = [x_2](k) + \tau_{1,2}$
5      $[w_3](k) = \min([w_1](k), [w_2](k))$
6      $[w_4](k) = [u_1](k) + \alpha_{1,1}$
7      $[x_1](k+1) = \max([w_3](k), [w_4](k))$
8      $[x_2](k+1) = [x_1](k) + \tau_{2,1}$
9      $[y_1](k) = [x_1](k) + \delta_{1,1}$
10      $[y_2](k) = [x_2](k) + \delta_{2,2}$
11   **end**
12   **for** $k=\overline{k}$ **downto** 0
13      $[z_1](k) = [z_1](k) \cap ([y_1](k) + [e_1](k))$
14      $[z_2](k) = [z_2](k) \cap ([y_2](k) + [e_2](k))$
15      $[y_1](k) = [y_1](k) \cap ([z_1](k) - [e_1](k))$
16      $[y_2](k) = [y_2](k) \cap ([z_2](k) - [e_2](k))$
17      $[x_2](k) = [x_2](k) \cap ([y_2](k) - \delta_{2,2})$
18      $[x_1](k) = [x_1](k) \cap ([y_1](k) - \delta_{1,1})$
19      $[x_1](k) = [x_1](k) \cap ([x_2](k+1) - \tau_{2,1})$
20   CMAX$([x_1](k+1), [w_3](k), [w_4](k))$
21      $[u_1](k) = [u_1](k) \cap ([w_4](k) - \alpha_{1,1})$
22   CMIN$([w_3](k), [w_1](k), [w_2](k))$
23      $[x_2](k) = [x_2](k) \cap ([w_2](k) - \tau_{1,2})$
24      $[x_1](k) = [x_1](k) \cap ([w_1](k) - \tau_{1,1})$
25   **end**
26   **while** contraction is significant

Table A.1. Forward-Backward Propagation for Problem 2

Algorithm CMIN (inout : $[z], [x], [y]$)
1   **if** $([z] \cap [x] = \emptyset)$ **then** $[y] = [y] \cap [z]$
2   **else if** $([z] \cap [y] = \emptyset)$ **then** $[x] = [x] \cap [z]$
3   $[x] = [x] \cap [z^-, \infty]$
4   $[y] = [y] \cap [z^-, \infty]$

Table A.2. Contractor for $[z] = \min([x], [y])$

REFERENCES

Ayhan, H. and Wortman, M. (1999). Job flow control in assembly operations. *IEEE Transactions on Automatic Control*, 44(4), 864–868.

Baccelli, F., Cohen, G., Olsder, G., and Quadrat, J. (1992). *Synchronization and Linearity. An Algebra for Discrete Event Systems*. John Wiley & Sons, New York.

Blyth, T. and Janowitz, M. (1972). *Residuation Theory*. Pergamon press.

Boudec, J.Y.L. and Thiran, P. (2002). *Network Calculus*. Springer Verlag.

Chabert, G. and Jaulin, L. (2009). Contractor programming. *Artificial Intelligence*, 173(11), 1079–1100. doi:10.1016/j.artint.2009.03.002. URL http://www.emn.fr/x-info/gchabe08/quimper.pdf.

Chen, W. and Tao, Y. (2001). Observabilities and reachabilities of nonlinear deds and coloring graphs. *Chinese Science Bulletin*, 46, 642–644.

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

De Schutter, B. and van den Boom, T. (2000). Model predictive control for max-min-plus systems. In *Discrete Event Systems: Analysis and Control* (Proceedings of the 5th International Workshop on Discrete Event Systems (WODES2000), Ghent, Belgium, Aug. 2000).

Gunawardena, J. (1994). Min-max functions. *Discrete Event Dynamic Systems*, 4, 377–406.

Gunawardena, J. and Keane, M. (1995). The existence of cycle times for some nonexpansive maps. Technical Report HPL-BRIMS-95-003, Hewlett-Packard Labs.

Jaulin, L., Ratschan, S., and Hardouin, L. (2004). Set computation for nonlinear control. *Reliable Computing*, 10(1), 1–26.

Maia, C.A., Hardouin, L., Santos-Mendes, R., and Cottenceau, B. (2003). Optimal closed-loop control of timed-event graphs in dioids. 48(12), 2284–2287. IEEE Transactions on Automatic Control.

Menguy, E., Boimond, J.L., Hardouin, L., and Ferrier, J.L. (2000). Just in Time Control of Timed Event Graphs: Update of Reference Input, Presence of Uncontrollable Input. *IEEE Trans. on Automatic Control*, 45(11), 2155–2158.

Montanari, U. (1974). Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences*, 7(66), 95–132.

Moore, R.E. (1979). *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, PA.

Murata, T. (1989). Petri nets : properties, analysis and applications. *Proceedings of the IEEE*, 77(4), 541–580.

Necoara, I., Kerrigan, E., De Schutter, B., and van den Boom, T. (2007). Finite-horizon min-max control of max-plus-linear systems. *IEEE Transactions on Automatic Control*, 52(6), 1088–1093. doi: 10.1109/TAC.2007.899071.

Rump, S.M. (1995). INTLAB: Interval laboratory, a MATLAB toolbox for interval arithmetic. World-Wide Web document. URL http://www.ti3.tu-harburg.de/rump/intlab.

Wen-De Chen, Yue-Gang Tao, H.N.Y. (2010). Independent cycle time assignment for min-max systems. *International Journal of Automation and Computing*, 7(2), 254.

Zhu, Y., Tao, Y., and Liu, G.P. (2009). Output feedback stabilization for a class of nonlinear time-evolution systems. *Nonlinear Analysis: Theory, Methods and Applications*, 70(7), 2654 – 2664. doi: 10.1016/j.na.2008.03.052.