Causality phenomenon and Compromise Technique for Predictive Control of Timed Event Graphs with Specifications Defined by P-time Event Graphs

Philippe Declerck *

* *LISA/LARIS EA7315, University of Angers, 62 avenue Notre-Dame du Lac, 49000 Angers, France (e-mail: philippe.declerck@univ-angers.fr).*

Abstract: An imperative condition of operation in Predictive Control is that a control must be applied after the end of its calculation. In this paper, we analyze and formalize this causality phenomenon which depends on both the computer time and the control problem. Two techniques are proposed. When the causality forbids the complete convergence of the algorithm, we propose a compromise technique which needs the analysis of the partial satisfaction of the specifications at each iteration of the algorithm. The plant is described by a Timed Event Graph while the specifications are defined by a P-time Event Graph.

Keywords: Timed Event Graphs, P-time Petri nets, (min, max, +) functions, fixed point, predictive control, causality.

1. INTRODUCTION

A classical problem is the control of a Timed Event Graph where some events are stated as controllable, meaning that the corresponding transitions (input) may be delayed from firing until some arbitrary time provided by a supervisor. In this paper, the specifications are defined by a P-time Event Graph which describes the desired behavior of the interconnections of all internal transitions. We wish to determine an input in order to obtain the desired behavior defined by the specifications.

This subject or a variant of this problem have already been considered in many papers but, the causality phenomenon which poses a problem does not seem to be fully considered at the best of our knowledge. Approaches based on a feedback defined by a Petri net are limited by the condition that the duration and the initial marking of each added place are non-negative. The existence of a linear state feedback is discussed in (Katz (2007)). In a similar way, the approaches based on a prediction (Model Predictive Control) present an analogous difficulty as they calculate a future control which must be applied on-line: for the above procedure, the application of the control must be made after the past dates of the state which are the known initial starting point of the problem. More precisely, the application of the first calculated control must be made after the addition of the last past date of the known state and the computer time. This difficulty arises if we consider the practical control of large scale systems as transportation systems, process with delays in the application of the control, real-time systems, etc.

Therefore, the aim of this paper is to deal with this causal constraint and to propose different techniques when a predictive control is used. A first objective is the analysis of the causality phenomenon and the determination of its effects on the control approach. In this paper, we will show that a possible technique (denoted technique 1 in this paper) is to modify the control such that the causal constraint is satisfied.

With the aim of fulfill the requirements of this time constraint, a natural objective is to improve the on-line procedure. A technique is to avoid the repetition of the same calculations at each iteration which can be costly in terms of time. Before the application of the on-line control, a *preparation* can contain these calculations allowing a reduction of the complexity of the on-line procedure (Declerck (2013)). Another technique (denoted technique 2) given in (Declerck and Guezzi (2012)) is based on a restriction of the state space leading to a convergence of the algorithm at the first iteration under a space condition.

However, the above approaches can be insufficient or do not succeed in the practical context of the control problem: we must also consider the case where no technique overcomes the causality problem. Therefore, we propose an approach based on a compromise approach whose main points are as follows. If a fixed point algorithm is used, we can reduce the CPU time by stopping the algorithm before the occurrence of a causality problem. The control generated by this unusual technique (denoted technique 3) is suboptimal as the convergence is not waited and only a subset of the constraints is satisfied. This approach can be sufficient if the important constraints are guaranteed by an analysis. Clearly, the satisfaction of safety regulations for a grade crossing is obligatory contrary to the following non-crucial constraint taken in the food industry: In good bakery practice, the dough stays in the fermentation room from three to five hours, the time depending on room temperature and flour or gluten quality; if these times are too short or too long, the quality of the product will slightly be damaged (bad inner structure and grain in the finished loaf). Therefore, the resolution of this problem implies that we focus on the validity of the constraints at each iteration which allows the application of suboptimal control to the process before the convergence of the algorithm.

The broad outline of the paper is as follows. Firstly, we describe the control problem and the fixed point algorithm which calculates the control and the state trajectories (Declerck and Guezzi (2012)). Then, we analyze the causality phenomenon in predictive control and propose Technique 1. Based on a basic theorem of (max, +) algebra, the consistency of the constraints is finally analyzed and a second Technique 3 is proposed. A pedagogical example using a variation of the model given in (Declerck and Guezzi (2012)) illustrates the main points. Due to lack of space, the preliminary remarks are also given in this reference.

2. CONTROL PROBLEM

Let us consider the initial control problem of this paper defined over $\mathbb{R}_{\max}$. Below, the variable $x_i(k)$ is the date of the k^{th} firing of the transition x_i and n is the dimension of $x(k)$. In this paper, we consider a classical predictive control based on the infinite repetition of the following control step on a sliding finite horizon. For one control step, the objective of this paper is the determination of the greatest control u (with respect to the componentwise order) on an arbitrary horizon $[k_s+1, k_f]$ with $h = k_f - k_s \in \mathbb{N}$ such that its application to the Timed Event Graph defined by

$$\begin{cases} x(k+1) = A \otimes x(k) \oplus B \otimes u(k+1) \\ y(k) = C \otimes x(k) \end{cases} \tag{1}$$

for $k \geq k_s$, satisfies the following conditions:

a) $y \leq \underline{z}$ knowing the trajectory of the desired output $\underline{z}$;

b) The state trajectory follows the model of the P-time Event Graph defined by

$$\begin{pmatrix} x(k) \\ x(k+1) \end{pmatrix} \geq \begin{pmatrix} A^{=} & A^{+} \\ A^{-} & A^{=} \end{pmatrix} \otimes \begin{pmatrix} x(k) \\ x(k+1) \end{pmatrix}; \tag{2}$$

c) The initial value of the state trajectory $x(k)$ for $k \geq k_s$ is finite and is a known vector denoted $\underline{x}(k_s)$. This " non-canonical " initial condition can be the result of a past evolution of a process. Since $\underline{x}(k_s)$ is finite, the trajectories considered in this paper are finite.

Underlined symbols like $\underline{x}(k_s)$ correspond to known data of the problem and $x(k)$ and $y(k)$ are estimated in the following resolutions.

The system (2) can always be obtained and corresponds to a P-time Event Graph where the initial marking of each place is equal to one at the greatest. When we consider the places having a unitary (respectively, null) initial marking, the lower bound a of the temporization of the place linking its ingoing transition x_j to its outgoing transition x_i generates the entry $A^{-}_{i,j} = a \geq 0$ (respectively, $A^{=}_{i,j} = a \geq 0$) and the upper bound b of the temporization of the place linking its ingoing transition x_i to its outgoing transition x_j generates the entry $A^{+}_{i,j} = -b \leq 0$ (respectively, $A^{=}_{i,j} = -b \leq 0$). More details can be found in (Declerck (2013)).

2.1 Relations on horizon $[k_s, k_f]$

The relations of the Timed Event Graph can be rewritten under the following classical form on horizon $[k_s, k_f]$.

$$X = \Omega_h \otimes x(k_s) \oplus \Psi_h \otimes U \tag{3}$$

where $h = k_f - k_s$,
$X = \left(x(k_s+1)^t \; x(k_s+2)^t \; \cdots \; x(k_f-1)^t \; x(k_f)^t \right)^t$ (t: transposed),
$U = \left(u(k_s+1)^t \; u(k_s+2)^t \; \cdots \; u(k_f-1)^t \; u(k_f)^t \right)^t$, Ω_h is a column of h blocks $(\Omega_h)_i = A^i$ for $i = 1$ to h and Ψ_h is a h x h matrix of blocks $(\Psi_h)_{i,j}$ for $i, j \in \{1, 2, \ldots, h\}$ where $(\Psi_h)_{i,j} = A^{i-j} \otimes B$ for $i > j$ and ε otherwise.

Below we consider the additional constraints (2) for $k \geq k_s$ and an autonomous Timed Event Graph defined by the inequality $x(k) \geq A \otimes x(k-1)$ which is the relaxation of the earliest firing rule, starting from $x(k_s) = \underline{x}(k_s)$.

$$\begin{cases} \begin{pmatrix} x(k_s) \\ X \end{pmatrix} \geq D_h \otimes \begin{pmatrix} x(k_s) \\ X \end{pmatrix} \\ x(k_s) = \underline{x}(k_s) \end{cases} \tag{4}$$

where D_h is a tridiagonal matrix of blocks $(D_h)_{i,j}$ for $i, j \in \{1, 2, \ldots, h+1\}$: This square matrix is composed of a main diagonal ($(D_h)_{i,i} = A^{=}$ for $i \in [1, h+1]$), an upper diagonal ($(D_h)_{i,i+1} = A^{+}$ for $i \in [1, h]$), a lower diagonal ($(D_h)_{j+1,j} = A \oplus A^{-}$ for $j \in [1, h]$); all other blocks are zero matrices (square submatrix ε). The matrix D_h is a $n.(h+1)$ x $n.(h+1)$ matrix where n is the dimension of x. More details can be found in (Declerck (2013)).

2.2 Fixed point form and algorithm

We introduce the following extended state vector $\overline{x} = \left((x(k_s))^t \; (X)^t \right)^t$ which expresses the complete state trajectory. Let $(\overline{x})^+$ be the greatest estimate of state trajectory and $F =$

$$\left(\underline{x}(k_s)^t \; (C \backslash \underline{z}(k_s+1))^t \; (C \backslash \underline{z}(k_s+2))^t \; \cdots \; (C \backslash \underline{z}(k_f))^t \right)^t .$$

Theorem 1. (Declerck and Guezzi (2012)) The greatest state and control trajectory of the control problem is the greatest solution of the following fixed point inequality system

$$\begin{cases} \overline{x} \leq D_h \backslash \overline{x} \wedge F \\ U \leq \Psi_h \backslash X \\ X \leq \Omega_h \otimes x(k_s) \oplus \Psi_h \otimes U \end{cases} \tag{5}$$

with condition $\underline{x}(k_s) \leq x^+(k_s)$. ■

The effective calculation of the greatest control can be made by the classical iterative algorithm of McMillan and Dill (1992). The general resolution of $x \leq f(x)$ is given by the iterations of $x_{\langle i \rangle} \leftarrow x_{\langle i-1 \rangle} \wedge f(x_{\langle i-1 \rangle})$ if the finite starting point $x_{\langle 0 \rangle}$ is greater than the final solution. Here, number $\langle i \rangle$ represents the number of iterations and not the number of components of vector x.

An algorithm specific to the determination of the greatest state and control is given below. Since it follows the algorithm of McMillan and Dill, this algorithm is also

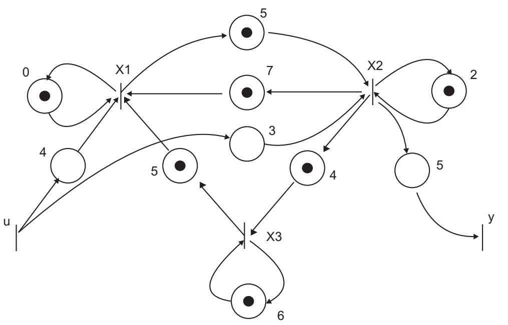

Fig. 1. Plant: Timed Event Graph (variation of the example given in (Declerck and Guezzi (2012))

pseudo-polynomial. Starting from $x_{\langle 0\rangle} = F$, the trajectory $\overline{x}$ is minimized in each iteration of the following algorithm where $(\overline{x})^1 = \left((x^1(k_s))^t \ (X^1)^t \right)^t$ and $(\overline{x})^2 = \left((x^2(k_s))^t \ (X^2)^t \right)^t$ correspond to useful intermediate values. Each iteration $\langle i\rangle$ with $i > 0$ considers the three steps 1, 2 and 3.

Algorithm 1 (Declerck and Guezzi (2012))

Step 0 (initialization): $\langle i\rangle \leftarrow \langle 0\rangle$; $(\overline{x})^2 \leftarrow F$

Repeat

- $\langle i\rangle \leftarrow \langle i+1\rangle$ (numbering of the iteration)
- Step 1: $(\overline{x})^1 \leftarrow D_h^* \backslash (\overline{x})^2$
- Step 2: $U \leftarrow \Psi_h \backslash X^1$
- Step 3: $(\overline{x})^2 \leftarrow (\overline{x})^1 \wedge \begin{pmatrix} +\infty \\ \Omega_h \otimes x^1(k_s) \oplus \Psi_h \otimes U \end{pmatrix}$

until $X^1 = X^2$. ■

Step 1 is deduced from the resolution of $\overline{x} \leq D_h \backslash \overline{x} \wedge (\overline{x})^2$. The obtained solution $(\overline{x})^1$naturally satisfies $(\overline{x})^1 \leq D_h \backslash (\overline{x})^1$ which is equivalent to the first relation in (4). The rest of the algorithm checks that this calculated solution, also satisfies $X^1 = \Omega_h \otimes x^1(k_s) \oplus \Psi_h \otimes U$ with $U = \Psi_h \backslash X^1$.

Used in section 4.2, the following result shows the minimisation of the state trajectory and the property that the state equation is satisfied at the end of each iteration.

***Property** 1.* (Declerck and Guezzi (2012)) $X' \leq X^1$ and $X^2 = X'$ where $X' = \Omega_h \otimes x^1(k_s) \oplus \Psi_h \otimes U$.

Algorithm 1 proposes an initial state $x^1(k_s)$ satisfying $x^1(k_s) \leq \underline{x}(k_s)$ and generates a trajectory starting from $x^1(k_s)$ given by the expression $\Omega_h \otimes x^1(k_s) \oplus \Psi_h \otimes U$.

2.3 Example 1

This example is a variation of the example given in (Declerck and Guezzi (2012)): We will show that the new matrix B leads to a more complex convergence.

Timed Event Graph (Fig. 1): $A = \begin{pmatrix} 0 & 7 & 5 \\ 5 & 2 & \varepsilon \\ \varepsilon & 4 & 6 \end{pmatrix}$, $B = (\, 4 \ 3 \ \varepsilon \,)^t$ and $C = (\, \varepsilon \ 5 \ \varepsilon \,)$

P-time Event Graph (Fig. 2): $A^{=} = \begin{pmatrix} \varepsilon & \varepsilon & -11 \\ \varepsilon & \varepsilon & -11 \\ 1 & 1 & \varepsilon \end{pmatrix}$,

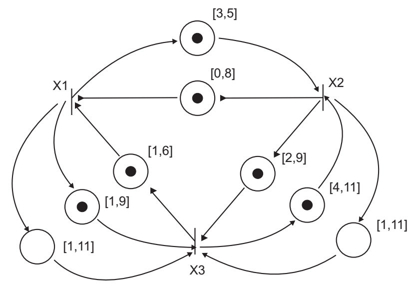

Fig. 2. Specifications: P-Time Event Graph (Declerck and Guezzi (2012))

$A^- = \begin{pmatrix} \varepsilon & 0 & 1 \\ 3 & \varepsilon & 4 \\ 1 & 2 & \varepsilon \end{pmatrix}$ and $A^+ = \begin{pmatrix} \varepsilon & -5 & -9 \\ -8 & \varepsilon & -9 \\ -6 & -11 & \varepsilon \end{pmatrix}$

Taking $h = 3$, the desired output $z(k)$ and the initial condition $\underline{x}(k_s)$ are as follows:
$z(k) = 25,\ 25,\ 28$ for $k_s + 1 \leq k \leq k_s + 3$ and $\underline{x}(k_s) = (\, 2 \ 0 \ 3 \,)^t$. Needing two iterations, Algorithm 1 gives the following results: $u(k) = 4,\ 10,\ 16$ for $k_s+1 \leq k \leq k_s+3$, $x(k_s) = (\, 2 \ 0 \ 3 \,)^t$, $x(k_s+1) = (\, 8 \ 7 \ 9 \,)^t$, $x(k_s+2) = (\, 14 \ 13 \ 15 \,)^t$, $x(k_s+3) = (\, 20 \ 19 \ 21 \,)^t$and $y(k) = 12,\ 18,\ 24$ for $k_s+1 \leq k \leq k_s+3$. The new matrix Ψ_h is given below. By lack of place, matrix $(D_h)^*$ is not given but can be found in (Declerck and Guezzi (2012)).

$$(\Psi_h)^t = \begin{pmatrix} 4 & 3 & \varepsilon & 10 & 9 & 7 & 16 & 15 & 13 \\ \varepsilon & \varepsilon & \varepsilon & 4 & 3 & \varepsilon & 10 & 9 & 7 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4 & 3 & \varepsilon \end{pmatrix}$$

3. CAUSALITY

3.1 Causality phenomenon and technique 1

Approaches based on a feedback defined by a Petri net are limited by the condition that the duration and the initial marking of each added place are non-negative. The existence of a linear state feedback is discussed in (Katz (2007)). In a similar way, the approaches based on a prediction (Model Predictive Control) present an analogous difficulty as they calculate a future control which must be applied on-line: for the above procedure, the application of the control $u(k_s+1)$ must be made after the dates of $\underline{x}(k_s)$ which are the data of the problem.

Therefore, each component $(u(k_s+1))_i$ must be greater than the date of the possible application which is the addition (in the standard algebra) of the maximum of the components of $\underline{x}(k_s)$ and the computer time T_{comp} which is the time taken from the start of the algorithm until the end as measured by an ordinary clock (more details are given in section 3.2). More formally, we have

$$\bigoplus_{i\in[1,n]} \underline{x}_i(k_s) \otimes T_{comp} \leq \bigwedge_{i\in[1,card(u)]} u_i(k_s+1) \qquad (6)$$

where $\bigoplus_{i\in[1,n]} \underline{x}_i(k_s) \otimes T_{comp}$ is the availability date of the calculated control. We can also rewrite this causality condition under the form of a (max, +) inequality

$$u(k_s+1) \geq G_u \otimes \underline{x}(k_s) \qquad (7)$$

where G_u is the $\otimes$−product of T_{comp} and a full matrix of zeros ($e = 0$) with appropriate dimensions. A control satisfying inequality (7) is called "causal". The algorithm can directly consider this causal constraint by adding $x(k_s) \leq G_u \backslash u(k_s + 1)$ in the constraints of the control problem which leads to a mimimization of the trajectories. Note that the introduction of the above constraints can naturally modify the trajectories and leads to a minimized control and output (another approach implying a maximized control will be proposed in a next paper). It can also change the consistency of the system.

3.2 CPU time and complexity of the on-line control

As the causality phenomenon is a constraint depending on the computer time, we consider the CPU time which gives a picture of the computer time T_{comp}: recall that the CPU time is the amount of time for which a central processing unit was used for processing instructions of a computer program contrary to the computer time T_{comp} which includes the CPU time and also the variable time spent by the computer in executing Kernel routines.

In our approach, the CPU time of the control does not include the calculation of the matrices Ω_h, Ψ_h and star $(D_h)^*$ which is made in the *off-line preparation* and depends on the matrices of the models and the size of the horizon. Therefore, the calculations of the on-line control are only limited to simple operations: a multiplication, an addition, a minimisation and two left $\otimes$−residuation of matrices. The complexity of one iteration of Algorithm 1 is $O(h^2.n.\max((n, q))$ with $q = card(u)$ (remember that $h = k_f - k_s$ and $n = card(x)$). We also made computation tests on one iteration of Algorithm 1 using the max-plus toolbox in Scilab 3.1.1 with an Intel Core2 Duo 2.26 GHz. For $n = 50$, the off-line preparation approximately needs 362 seconds while the on-line procedure only needs 0.12 seconds. Therefore, the computer time T_{comp} of the on-line control is drastically reduced by the *off-line preparation.*

4. CONSTRAINT CONSISTENCY

As the objective of Algorithm 1 is to fulfil the requirements of the control problem, we now focus on the satisfaction of the different constraints at each iteration which can be complete or partial. Before the description of the relevant studies, we present Theorem 2 which poses a general problem and highlights an important case where Algorithm 1 gives the final state trajectory at the first iteration $\langle 1\rangle$: It implies that Algorithm 1 is strongly polynomial since the resolution is reduced to a unique iteration composed of the simple application of elementary operations $\oplus$, $\otimes$, $\wedge$ and $\backslash$.

***Theorem** 2.* (Declerck and Guezzi (2012)) The trajectory $(\overline{x})^2$ satisfies the system composed of (3) and the first relation of (4) when $\begin{pmatrix} I & \varepsilon \\ \varepsilon & \Psi_h \end{pmatrix} \otimes \begin{pmatrix} x^0(k_s) \\ U \end{pmatrix} = (\overline{x})^1$. Moreover, $(\overline{x})^2 = (\overline{x})^1$. ■

Rewritten with a simpler notation, the condition of Theorem 2 is analyzed in the following sections: The problem is to check the solution existence of $\overline{u} \in \mathbb{R}^{\overline{q}}$ in the equality

$$\overline{B} \otimes \overline{u} = \overline{x} \text{ for any } \overline{x} \in \mathbb{R}^{\overline{n}} \text{ satisfying } \overline{x} \geq \overline{A} \otimes \overline{x} \qquad (8)$$

with the following notation: $\overline{B} = \begin{pmatrix} I & \varepsilon \\ \varepsilon & \Psi_h \end{pmatrix}$, $\overline{u} = \begin{pmatrix} x(k_s) \\ U \end{pmatrix}$, $\overline{x} = \begin{pmatrix} x(k_s) \\ X \end{pmatrix}$, $\overline{A} = D_h$, $\overline{n}$=card($\overline{x}$) and $\overline{q}$=$card(\overline{u})$. We assume in (8) that matrix $\overline{B}$ has no null rows as $\overline{x}$ is finite. Without a loss of generality, we assume that matrix $\overline{B}$ has no null columns so that $\overline{u}$=$\overline{B} \backslash \overline{x}$ is finite. We naturally assume that the associated graph of $\overline{A}$ does not contain circuits with strictly positive weight so that $\overline{A}^* \in \mathbb{R}_{\max}^{\overline{n}x\overline{n}}$. The objective is also to obtain practical tests which use only the entries of $\overline{B}$ and $\overline{A}^*$ without calculating the state and the control

4.1 Complete validity of the constraints and space controller (technique 2) (Declerck and Guezzi (2012))

Property 2 analyzes the existence of a solution $\overline{u}$ in (8).

***Property** 2.* (Declerck and Guezzi (2012)) The greatest vector $\overline{u}$=$\overline{B} \backslash \overline{x}$ satisfies the system (8) if and only if $\overline{B} \otimes (\overline{B} \backslash \overline{A}^*) = \overline{A}^*$. ■

Therefore, Property 2 provides conditions described by Theorem 2 which lead to a convergence in one iteration. When $\overline{B} \otimes (\overline{B} \backslash \overline{A}^*) = \overline{A}^*$ is not satisfied for all columns $(\overline{A}^*)_{.,k}$ but for some columns denoted $(\overline{A}^*)^{=}_{.,k}$, a property in (Declerck and Guezzi (2012)) shows that Algorithm 1 can stop for any iteration when $\overline{x} \in \mathrm{Im}(\overline{A}^*)^{=}$. Since the fulfilment of this condition is not guaranteed, we proposed a predictive control using a space controller in (Declerck and Guezzi (2012)). This approach compensates for the non-satisfaction of the condition $\overline{B} \otimes (\overline{B} \backslash \overline{A}^*) = \overline{A}^*$ by reducing the state space to the subspace $\mathrm{Im}(\overline{A}^*)^{=}$ under the condition $\mathrm{Im}(\overline{A}^*)^{=} \neq \emptyset$. The relevant algorithm is strongly polynomial, contrary to Algorithm 1 considered without additionnal condition. However, Algorithm 1 can still be used even if $\mathrm{Im}(\overline{A}^*)^{=} = \emptyset$.

4.2 Partial validity of the constraints and compromise technique (technique 3)

In the previous section, we consider an important case where the convergence of Algorithm 1 is efficient as it is reduced to one iteration. The conditions of this important case can be fullfilled by a space controller based on a restriction of the space.

We now consider that the above techniques cannot be applied. The addition of (7) in the constraints is not possible or not desired because it leads to a control with a major delay. We cannot apply a predictive control using a space controller as the conditions are not satisfied. More precisely, we cannot reduce the CPU time by applying a predictive control using a space controller because $\mathrm{Im}(\overline{A}^*)^{=} = \emptyset$. Moreover, we consider the case where the control calculated by Algorithm 1 is not causal.

Therefore, we consider in this section Algorithm 1 without the addition of (7) and analyze the consistency of each row of system (8): The mathematical objective is to generalize Theorem 2 and Property 2.

As we consider that the control cannot be postponed, we now propose a symmetrical approach which is to reduce the CPU time by stopping Algorithm 1 before the convergence such that inequality (7) is still satisfied for the current calculated control. This unusual technique is realistic as the algorithm makes a minimization of the variables: each iteration of the algorithm proposes a control which generates an output satisfying the desired output (expressed by vector F) and the state equation (Property 1). Moreover, a subset of the constraints is satisfied. This suboptimal solution can be sufficient if we can guarantee the important constraints such as safety regulations. Naturally, the maximal number of iterations under the causality condition must be taken as the solution is improved at each iteration (except at the end of the convergence which check the stability of the solution). Therefore, a compromise must be made between:

- The **increase** of the availability date expressed by the right hand term of the causal condition (7) (and produced by the increase of the computer time which depends on the number of iterations)
- and, the **decrease** of the dates of the control which is the left hand term in (7) (and is minimized by the fixed point algorithm at each iteration) .

We now analyze the validity of the constraints at each iteration of Algorithm 1. Contrary to the previous section, the validity can be partial. The approach uses the following lemma which considers the finite solution to $A \otimes x = b$ where $A \in \mathbb{R}_{\max}^{mxn}$, $b \in \mathbb{R}^m$. The relevant set of solutions over $\mathbb{R}$ is denoted $\mathcal{S}$. This lemma is a slight generalization of Theorem 2.2 in (Butkovic and Tam (2009)) (R.A. Cuninghame-Green, K. Zimmermann, P. Butkovic) where A is over $\mathbb{R}$.

We denote the set of indexes for the rows $I = \{1,..,m\}$ and for the columns $J = \{1,..,n\}$ as A is a (mxn) matrix. Remember that x^+ is the greatest solution to $A \otimes x \leq b$. We consider the finite entries of A which can imply the equality $A_{i,j} \otimes x_j^+ = b_i$ where $A_{i,j}$, $x_j^+$and $b_i \in \mathbb{R}$: for $j \in J$, $V_j = \{i \in I$ such that $A_{i,j}$ is finite and $x_j^+ = A_{i,j} \backslash b_i\}$.

***Lemma** 1.* Let $A \in \mathbb{R}_{\max}^{mxn}$, $b \in \mathbb{R}^m$. So, $x \in \mathcal{S}$ if and only if $x \leq x^+$ and $\bigcup\limits_{j \in J \mid\ x_j = x_j^+} V_j = I$. ∎

***Corollary** 1.* The following three statements are equivalent: 1) card$(S) \neq 0$ 2) $x^+ \in S$ 3) $\bigcup\limits_{j \in J} V_j = I$. ∎

So, if a row of A is null, $\bigcup\limits_{j \in J \mid\ x_j = x_j^+} V_j \neq I$ and there is no finite solution (Moreover, there is no infinite solution as ε is absorbing and $b_j \in \mathbb{R}$). If a column A_j is null, V_j is empty and there is no effect on the equality.

The consideration of the following equality

$$\overline{B} \otimes \overline{v} = \overline{A}^* \ , \tag{9}$$

where $\overline{v}$ is a $(\overline{q}\mathrm{x}\overline{n})$ matrix will be useful. The greatest solution is denoted $\overline{v}^+$. Lemma 1 allows an analysis of each row of system (9) by inspection of the sets $V_{j,k}$ defined as follows. We denote the set of indexes for the rows $I = \{1,..,\overline{n}\}$ and for the columns $J = \{1,..,\overline{q}\}$ as $\overline{B}$ is a $(\overline{n}\mathrm{x}\overline{q})$ matrix. Let $K = \{1,..,\overline{n}\}$ be the set of indices of columns of $\overline{A}^*$. Corresponding to column $j \in J$ of $\overline{B}$ and column $k \in K$ of $\overline{A}^*$, $V_{j,k}$ is defined by $V_{j,k} = \{i \in I$ such that $\overline{B}_{i,j}$ is finite and $\overline{v}_{j,k}^+ = \overline{B}_{\ i,j} \backslash (\overline{A}^*)_{i,k}\}$.

***Property** 3.* The system (9) has a solution $\overline{v}$ if and only if $\overline{v} \leq \overline{v}^+$ and $\bigcap\limits_{k \in K} \bigcup\limits_{j \mid\ \mathbf{v}_{j,k} = \mathbf{v}_{j,k}^+} V_{j,k} = I$. The set $\bigcap\limits_{k \in K} \bigcup\limits_{j \in J} V_{j,k}$ gives the rows of (9) where the equality holds for $\overline{v}=\overline{v}^+$. ∎

Now we can consider the system (8). The following result generalizes the converse of property 2 (Section 4.1) by considering the consistency of each row.

***Property** 4.* For the greatest vector $\overline{u}=\overline{B} \setminus \overline{x}$,

- each equality $\overline{B}_{\ i,.} \otimes \overline{u} = \overline{x}_i$ with $i \in I_g = \bigcap\limits_{k \in K} \bigcup\limits_{j \in J} V_{j,k}$ is always satisfied for any $\overline{x} \in \mathrm{Im}\overline{A}^*$.
- each equality $\overline{B}_{\ i,.} \otimes \overline{u} = \overline{x}_i$ with $i \in I_{p,k} = \bigcup\limits_{j \in J} V_{j,k}$ is always satisfied when $\overline{x} \in \mathrm{Im}(\overline{A}^*)_{.,k}$ for a given $k \in K$.
- An equality $\overline{B}_{\ i,.} \otimes \overline{u} = \overline{x}_i$ with $i \in I_p = \{i \in \bigcup\limits_{k \in K} \bigcup\limits_{j \in J} V_{j,k}$ and $i \notin I_g\}$ is possibly satisfied when $\overline{x} \in \mathrm{Im}\overline{A}^*$. ∎

In the first point, the set I_g guarantees the consistency of a subset of constraints in (9) for any state trajectory $\overline{x} \in \mathrm{Im}\overline{A}^*$. The same remark holds for the set $I_{p,k}$ but the state trajectory $\overline{x}$ follows a unique direction $(\overline{A}^*)_{.,k}$ with $k \in K : \overline{x} = \lambda_k \otimes (\overline{A}^*)_{.,k}$. We directly deduce that, depending of the state evolution inside the space $\mathrm{Im}\overline{A}^*$, the set I_p gives the rows of $\overline{B} \otimes \overline{u} = \overline{x}$ where the equality is possibly satisfied.

Considering system (4) (Section 2.1) at each iteration of Algorithm 1, the following theorem generalizes Theorem 2.

***Theorem** 3.* The constraints corresponding to the rows $i \in I_g$ of the first relations of (4) are satisfied at each iteration of Algorithm 1 for the control calculated at step 2. ∎

When there is no restriction produced by the causality, Algorithm 1 must be continued until the convergence where all the constraints are satisfied. If a causality problem occurs, Algorithm must be stopped before its normal convergence: Therefore, we can guarantee the satisfaction of a subset of the constraints at the end of the last executed iteration (at the end of step 3) for the control calculated at step 2.

Remark 1. The situation $I \neq I_g$ only means that the conditions of Property 2 (Section 4.1) leading to a convergence in one iteration are not satisfied. It does not imply that the general control problem is inconsistent as it also depends on the initial state $x(k_s)$. In fact, some rows of $\overline{B}_{\ i,.} \otimes \overline{u} = \overline{x}_i$ can be inconsistent. See below Example 1.

4.3 Example 1 (continued).

We have $\overline{n} = (h+1).n = 12$ and $\overline{q} = n + h.card(u) = 6$ as $n = 3$ and $h = 3$. So, $I = \{1,...,12\}$, $J = \{1,...,6\}$ and $K = \{1,...,12\}$. Remember that $\overline{A}^* = (D_h)^*$ and

$\overline{B} = \begin{pmatrix} I & \varepsilon \\ \varepsilon & \Psi_h \end{pmatrix}$. Each entry $\Delta_{i,k}$ of the following $\overline{n}$ x $\overline{n}$ symbol matrix gives the row index $i \in \bigcup_{j \in J} V_{j,k}$ for each column $(\overline{A}^*)_{.,k}$ where symbol $=$ expresses that the relevant equality $\overline{B}_{i,.} \otimes \overline{v}_{.,k} = (\overline{A}^*)_{i,k}$ is satisfied while symbol $<$ shows that $\overline{B}_{i,.} \otimes \overline{v}_{.,k} < (\overline{A}^*)_{i,k}$ is obtained.

$$\Delta = \begin{pmatrix} = & = & = & = & = & = & = & = & = & = & = & = \\ = & = & = & = & = & = & = & = & = & = & = & = \\ = & = & = & = & = & = & = & = & = & = & = & = \\ = & < & = & < & = & < & = & < & = & < & = & < \\ = & = & = & = & = & = & = & = & = & = & = & = \\ < & < & < & < & < & < & < & < & < & < & < & < \\ = & = & = & = & = & = & < & = & < & = & < & = \\ = & = & = & = & = & = & = & = & = & = & = & = \\ < & < & < & < & < & < & < & < & < & < & < & < \\ = & = & = & = & = & = & = & = & = & < & = & = \\ = & = & = & = & = & = & = & = & = & = & = & = \\ < & < & < & < & < & < & < & < & < & < & < & < \end{pmatrix} \quad (10)$$

The analysis of the rows of this matrix Δ gives $I_g = \{1, 2, 3, 5, 8, 11\}$ and $I_p = \{4, 7, 10\}$.

Therefore, the equality $\overline{B} \otimes \overline{u} = (\overline{A}^*)_{.,k}$ does not hold for any $k \in K$ but the equality for the rows $i \in I_g$ is guaranteed. The constraints corresponding to $x_2(k_s+1)$, $x_2(k_s+2)$ and $x_2(k_s+3)$ are always satisfied, that is $x_2(k) \geq (A \oplus A^-)_{2,.} \otimes x(k-1) \oplus A^=_{2,.} \otimes x(k) \oplus A^+_{2,.} \otimes x(k+1)$ for $k = k_s + 1$ and $k_s + 2$ and $x_2(k) \geq (A \oplus A^-)_{2,.} \otimes x(k-1) \oplus A^=_{2,.} \otimes x(k)$ for $k = k_s + 3$. Remember that the equalities relevant to rows $\{1, 2, 3\}$ are always satisfied for any problem by construction of system (8). The following table generated by the simulation is coherent with these first results. Using the data obtained at the end of an iteration expressed by $(\overline{x})^2$, the table shows the evolution of the validity of the different constraints at each iteration, that is, the first relations of (4) (notation: g guaranteed; s satisfied; ns non satisfied). Only two iterations are necessary.

Iteration $\langle i \rangle$	$x_3(k_s{+}2)$	$x_1(k_s{+}3)$	**$x_2(k_s{+}3)$: g**	$x_3(k_s{+}3)$
1	15 (s)	21 (s)	20 (s)	21 (ns)
2	15 (s)	20 (s)	19 (s)	21 (s)

...

Iteration $\langle i \rangle$	$x_3(k_s{+}2)$	$x_1(k_s{+}3)$	**$x_2(k_s{+}3)$: g**	$x_3(k_s{+}3)$
1	15 (s)	21 (s)	20 (s)	21 (ns)
2	15 (s)	20 (s)	19 (s)	21 (s)

Relations relevant to $x_1(k_s + 2)$ and $x_3(k_s + 3)$ (which corresponds to rows 7 and 12 of $\overline{B}_{i,.} \otimes \overline{v} = \overline{A}^*$) at the end of iteration $\langle 1 \rangle$ are not satisfied: $x_1(k_s+2) = 14 \ngeq A^+_{1,2} \otimes x_2(k_s+3) = (-5) \otimes 20$ and $x_3(k_s+3) = 21 \ngeq A^=_{3,1} \otimes x_1(k_s+3) = 1 \otimes 21$.

Moreover, each equality $\overline{B}_i \otimes \overline{u} = \overline{x}_i$ for $i \in I_p = \{4, 7, 10\}$ is possibly satisfied. In the numerical example, all the constraints and $\overline{B}_i \otimes \overline{u} = \overline{x}_i$ for $i \in I_p = \{4, 7, 10\}$ are satisfied at the end of iteration $\langle 2 \rangle$.

Finally, the equalities $\overline{B}_i \otimes \overline{u} = \overline{x}_i$ for $i \in I_n = \{6, 9, 12\}$ are not necessary in this simulation. They are not satisfied at the end of iterations $\langle 1 \rangle$ and $\langle 2 \rangle$ (remark 1) but the equality of the state equation is obtained with $x_i(k) = A_{i,.} \otimes x(k-1)$: we have at the end of iteration $\langle 2 \rangle$:

$x_3(k) = A_{3,.} \otimes x(k-1)$ for $k = k_s + 1$, $k_s + 2$ and $k_s + 3$

$x_3(k_s+1) = 9 = A_{3,3} \otimes x_3(k_s) = 6 \otimes 3 > B_{3,.} \otimes u(k_s+1) = 4 \otimes 4$

$x_3(k_s+2) = 15 = A_{3,3} \otimes x_2(k_s+1) = 6 \otimes 9 > B_{3,.} \otimes u(k_s+2) = 4 \otimes 10$

$x_3(k_s+3) = 21 = A_{3,3} \otimes x_2(k_s+2) = 6 \otimes 15 > B_{3,.} \otimes u(k_s+3) = 4 \otimes 16$

5. CONCLUSION

In this paper, we analyze the causality phenomenon and enlarges the class of the processes where the predictive control can operate when the causality phenomenon forbids the application of the calculated control. Indeed, the computer can be too slow with respect to the size, the type and the data of the control problem. A first approach is to consider this limitation as a standard additional constraint which can be written in the (max, +) algebra. The proposed Technique 1 adds a minimization of the state and control trajectories under the condition of consistency of the modified system. Contrary to Techniques 1 and 2 (Declerck and Guezzi (2012)) where all the additional constraints must be satisfied, the second proposed Technique 3 can be applied when only a subset of crucial constraints must be satisfied. Moreover, we consider the situation where the conditions of the space controller (Technique 2) leading to a convergence in one iteration cannot be fulfilled. The suboptimal solution is the result of a compromise between the availability time of application of the control and the calculated dates. The analysis based on Lemma 1 has shown that a subset of the constraints are guaranteed while another subset is possibly satisfied.

REFERENCES

Butkovic P. and Tam K.P. (2009) On some properties of the image set of a max-linear mapping, Contemporary Mathematics Series, AMS Providence, 495, 115-126.

Declerck P. and Didi Alaoui M.K. (2010) Optimal control synthesis of timed event graphs with interval model specifications. Available from http://www.istia.univ-angers.fr/∼declerck, IEEE Transactions on Automatic Control, Vol.55, No.2, IETAA9, pp. 518-523, February.

Declerck P. (2013) Discrete Event Systems in Dioid Algebra and Conventional Algebra, Focus Series in Automation & Control, ISTE Ltd and John Wiley.

Declerck P. and Guezzi A. (2012) Predictive Control of Timed Event Graphs with Specifications Defined by P-time Event Graphs. Available from http://www.istia.univ-angers.fr/∼declerck, Journal of Discrete Event Dynamic Systems.

Katz R. D. (2007) Max-Plus (A,B)-Invariant Spaces and Control of Timed Discrete-Event Systems. IEEE Transactions on Automatic Control, Vol.52, No. 2, pp. 229-241.

McMillan K. and Dill D. (1992) Algorithms for interface timing verification. Proceedings of the IEEE, International Conference on Computer Design: VLSI in Computers and Processors.