# Trajectory Tracking Control of a Timed Event Graph with Specifications Defined by a P-time Event Graph: On-line control and Off-line preparation.

**Philippe Declerck** [*]

[*] *LISA EA4014, University of Angers, 62 avenue Notre-Dame du Lac, 49000 Angers, France (e-mail: philippe.declerck@univ-angers.fr).*

**Abstract:** The topic of this paper is the on-line trajectory tracking control on a sliding horizon of Timed Event Graphs with specifications defined by a P-time Event Graph. Making the most of the specific structures of the systems, the CPU time of the on-line procedure is drastically reduced with two techniques: The use of specific algorithms of the graph theory instead of generic algorithms; an off-line preparation which avoid the useless repetition of the same calculations. As a consequence, the approach can deal with long horizons and important systems and the application field of the trajectory tracking control is expanded.

*Keywords:* Timed Event Graph, P-time Petri Nets, Model predictive control, (max,+) algebra, large scale systems.

## 1. INTRODUCTION

In this paper, we focus on the control problem of Timed Event Graphs defined as follows. In a Timed Event Graph, 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. The specifications are defined by a P-time Event Graph which describes the desired behavior of the interconnections of all the internal transitions. P-time Event Graphs concern time-constrained systems where the duration of each operation is bounded by minimal and maximal limits. Contrary to the usual rule applied to Timed Event Graphs, the firing of the transitions is not "as soon as possible" otherwise some tokens can die which can lead to a deadlock situation. The token deaths express the losses of resources or parts and the failures to meet the time specifications. Applications of P-time Event Graphs can be found in production systems, microcircuit design, transportation systems and food industry.

The aim of this paper is to develop a model predictive control. Naturally, this topic has already been considered in different papers (Schutter et al. (2001)) where a usual step is to transform the (max, +) problem in a linear programming problem in the conventional algebra which allows the application of classical algorithms. The principal advantage of this technique is the consideration of different classes of models. However, model predictive control is an on-line approach which needs *efficient algorithms*: indeed, a crucial point is that a too slow calculation of the control can postpone the application of the control at the calculated dates.

The ELCP algorithm (Extended Linear Complementary Problem) described in chapter 3 of the thesis of Bart de Schutter cannot be used for on-line computations as the CPU time increases exponentially (Schutter et al. (2001)). The algorithms of Khashiyan and Karmarkar in linear programming are famous but it is well-known that they are polynomial *in the weak sense* (Savard (2001)) and not in the strong sense (contrary to many algorithms in graph theory: see chapter 2 in (Gondran et al. (1984))). The complexity of the Simplex is exponential in the worst case even if this algorithm is relatively good on the average (Pan (1985)). As a consequence, the application of these *generic* algorithms of linear programming (see the algorithms quoted in (Schutter et al. (2001))) leads to the limitation of the size of the considered systems and the magnitude of the entries. Another difficulty is that these generic algorithms correspond to the optimisation stage which need to start from an admissible solution which is the result of another stage. Not immediate, this determination can be included in the program (function simplex "linpro()" of Scilab) or not (function "karmarkar(a,b,c,x0)" of Scilab where x0 is the initial vector ). Therefore, a problem is the determination of an admissible solution close to the optimal solution.

In fact, the crucial point is that the structure of the matrices considered above present *specific characteristics*: the matrices are sparse and contains many rows with two non-null entries (1 and -1) at the most. The matrices are close to the ingoing/outgoing incidence matrices of the fundamental marking relation (Guezzi et al. (2008)). Therefore, the goal of the paper is to make the most of these specific structures of the systems and to deduce an approach having a reduced CPU time.

A general answer is to use the (max, +) algebra which allows the application of efficient algorithms of path theory: in general, they are *strongly* polynomial (the running time does not depend on the magnitudes of the parameters). A direct consequence is that these algorithms specific to path algebra, surpass the best generic algorithms of linear programming when they are applied to the specific problems: these algorithms can consider large scale systems. Our tests shows that, if the simplex is used to calculate the maximal paths between every pair of vertices (function "linpro()"), the approach can work until 60 vertices and need about 2 minutes while the (max, +) kleene star (function "star()") solves the problem in 1 seconde for 300 vertices. The difference generic/specific is also illustrated by a topical trend in computer science which consider linear inequalities with two inequalities per inequality (Cohen et al. (1994)) (Hochbaum (2004)) (Andersson et al. (2006)). This topic is also present in mathematics: A pioneer is G.B. Dantzig who analyses dynamic Leontief System in 1955; Cottle et al. (1972) has shown a correspondence between linear inequalities and lattices; A recent research is (Queyranne et al. (2006)).

Even if the use of the (max, +) algebra allows the application of fast algorithms (Declerck et al. (2009)), we propose an improvement of this technique in this paper. The result is a more formal (less numerical) approach by respect to the initial approach (Declerck et al. (2009)) and the extension of the predictive control to larger systems and horizons. In fact, the predictive control is made on a sliding horizon: the horizon is slightly moved back at each step and the control is calculated. The idea is to avoid the repetition of the same calculations at each step 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. In fact, the calculation of the kleene star of a tri-diagonal matrix can be made once as *it does not depend on the desired output trajectory*: it depends on the models and the size of the given horizon only. Note that the (max, +) algebra allows the writing of elegant expressions contrary to linear programming which can only give the numerical results. In this paper, we show that this separation between on-line preparation and off-line control where an important part of the calculations is made, allows the consideration of important systems (until 97 transitions) for long horizons ($h$ =50) which corresponds to the handling of a large scale matrix (4947x 4947). In that case, the initial CPU time (which needs approximately $2.10^3$ secondes) is replaced by a new on-line procedure which only needs 0.28 secondes. Therefore, the application field of the model predictive control is expanded.

A preparation is also made in closed-loop approaches where: the computation of a linear feedback and of the maximal set of the initial states is made off-line (Katz (2007)); the control is calculated on-line. Recall that approaches based on a feedback defined by a Petri net are limited by the condition that the temporisation and initial marking of each added place are non-negative. Our practical problem is different as the initial condition is uncontrollable which is the usual assumption of the predictive control approaches. Based on the algorithm of Butkovic et al. (1984) whose complexity is doubly exponential, the off-line preparation of Katz (2007) can only be made for small systems and an objective is the improvement of the algorithm. Another difficulty is the stabilization of the sequence of semi-modules and some sufficient conditions on the specifications are given (sections III and IV in (Katz (2007))). These difficulties are avoided in this paper as we define a subspace which is an inf-semilattice.

In this paper, we consider that each transition is observable: the event date of each transition firing is assumed to be available. Let us note that we have developed software written in Scilab composed of estimation, prediction and control. No hypothesis is taken on the structure of the Event Graphs which does not need to be strongly connected. The initial marking should only satisfy the classical liveness condition and the usual hypothesis that places should be First In First Out (FIFO) is taken.

The paper is structured as follows: In the first part, we consider the control problem without desired output and with a fixed horizon and the determination of the earliest trajectory (problem 1). In the following part, the desired output defined on a fixed horizon is introduced and a just-in-time control is made. The problem (problem 2) is to determine the greatest input in order to obtain a desired behavior defined by the desired output and the specifications. In the last part, the approach is generalized to a sliding horizon (problem 3) and we focus on the CPU time of the on-line procedure based on a preparation which calculates the kleene star of a large scale tri-diagonal matrix. By reason of the lack of space, the presentation of the P-time Event Graph and the analysis of the causality are omitted: the reader can find the presentation of the model in (Declerck (2007)) and an analysis of the causality in (Declerck et al. (2009)). We now give the notations.

Maximization, minimization and addition operations are denoted respectively $\oplus$, $\wedge$ and $\otimes$. The set of $n.n$ matrices with entries in dioid $D = \overline{\mathbb{R}}_{max} = (\mathbb{R} \cup \{-\infty\} \cup \{+\infty\}, \oplus, \otimes)$ including the two operations $\oplus$ and $\otimes$ is a dioid, which is denoted $D^{n.n}$. Mapping $f$ is said to be residuated if for all $y \in D$, the least upper bound of subset $\{x \in D \mid f(x) \leq y\}$ exists and lies in this subset. Mapping $x \in (\overline{\mathbb{R}}_{max})^n \mapsto A \otimes x$, defined over $\overline{\mathbb{R}}_{max}$ is residuated (see Baccelli et al. (1992)) and the left $\otimes-$residuation of $B$ by $A$ is denoted by: $A \backslash B = \max\{x \in (\overline{\mathbb{R}}_{max})^n$ such that $A \otimes x \leqslant B\}$. The Kleene star is defined by: $A^* = \bigoplus_{i=0}^{+\infty} A^i$. A matrix is called row-astic if it has no null row. Variable $x_i(k)$ is below the date of the $k^{th}$ firing of transition $x_i$ and $n$ is the dimension of $x$.

## 2. CONTROL WITHOUT DESIRED OUTPUT (PROBLEM 1)

### 2.1 Objective

The problem of this part is the control of a plant described by a Timed Event Graph when the state and control trajectories are constrained by additional specifications defined by a P-time Event graph. The objective is the determination of an admissible (arbitrary) control $u$ on horizon $[k_s + 1, k_f]$ such that its application to the Timed Event Graph defined by

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

satisfies the following conditions:

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

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

for $k \geq k_s$; matrix $A^-$ (respectively, $A^+$) contains the lower bounds of the temporizations (respectively, the upper bounds with minus sign) associated with each place having an unitary initial marking. We have $A^{=} = A_0^- \oplus A_0^+$ where matrix $A_0^-$ (respectively, $A_0^+$) is defined as $A^-$ (respectively, $A^+$) but with with a null initial marking.

b) The first state vector 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. As $x(k_s)$ 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.

Therefore, we focus on a control problem without desired output on a fixed horizon. This problem is denoted Problem 1.

The following example shows that the space solution to Problem 1 is not an inf-semilattice that is, Problem 1 has no (unique) minimal solution by respect to the componentwise order.

*Example 1* Timed Event Graph:

$A = \begin{pmatrix} T_1 & \varepsilon \\ T_2 & T_3 \end{pmatrix}$, $B = \begin{pmatrix} T_4 & T_5 & \varepsilon \\ \varepsilon & T_6 & T_7 \end{pmatrix}$ and $C = (\, \varepsilon \ T_8 \,)$ .

P-Time Event Graph:

$A^{=} = \varepsilon$, $A^{-} = \begin{pmatrix} T_{10} & T_{13} \\ T_{11} & T_{12} \end{pmatrix}$ and $A^{+} = \begin{pmatrix} -T_{20} & -T_{21} \\ -T_{23} & -T_{22} \end{pmatrix}$ .
Every temporisation is finite except $T_{20} = T_{21} = T_{22} = T_{23} = +\infty$ .

For $k_f = k_s + 1$, we have $x(k_s+1) = A \otimes \underline{x}(k_s) \oplus B \otimes u(k+1) \geq A^- \otimes \underline{x}(k_s)$. For $\underline{x}(k_s) = (\, 0 \ 0 \,)^t$,
$$\begin{cases} x_1(k_s+1) = T_1 \oplus T_4 \otimes u_1(k_s+1) \oplus T_5 \otimes u_2(k_s+1) \\ \geq T_{10} \oplus T_{13} \\ x_2(k_s+1) = T_2 \oplus T_3 \oplus T_6 \otimes u_2(k_s+1) \oplus T_7 \otimes u_3(k_s+1) \\ \geq T_{11} \oplus T_{12} \end{cases}$$
If $T_1 < T_{10} \oplus T_{13}$ and $T_2 \oplus T_3 < T_{11} \oplus T_{12}$, the system becomes:

$\begin{cases} T_4 \otimes u_1(k_s+1) \oplus T_5 \otimes u_2(k_s+1) \geq T_{10} \oplus T_{13} \\ T_6 \otimes u_2(k_s+1) \oplus T_7 \otimes u_3(k_s+1) \geq T_{11} \oplus T_{12} \end{cases}$ . This system has no (unique) minimal solution by respect to the componentwise order: if $u_2(k_s+1) = \varepsilon$, $u_1(k_s+1)$ and $u_3(k_s+1)$ must be finite; if $u_1(k_s+1) = \varepsilon$ and $u_3(k_s+1) = \varepsilon$, $u_2(k_s+1)$ must be finite. ■

In the following part 2.2, we present the relations which describe a trajectory of a Timed Event Graph satisfying the specifications defined by a P-time Event Graph (constraint a)).

### *2.2 Trajectory description*

From (1) and (2), we deduce a system which describes the trajectories on horizon $[k_s, k_f]$ . Let us introduce the following notations. Let $X = \big( x(k_s)^t \ x(k_s+1)^t \ x(k_s+2)^t \ \cdots \ x(k_f-1)^t \ x(k_f)^t \big)^t$ ($t$: transposed) and $D_h =$

$$\begin{pmatrix} A^{=} & A^{+} & \varepsilon & \cdots & \varepsilon & \varepsilon & \varepsilon \\ A \oplus A^{-} & A^{=} & A^{+} & \cdots & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & A \oplus A^{-} & A^{=} & \cdots & \varepsilon & \varepsilon & \varepsilon \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \varepsilon & \varepsilon & \varepsilon & \cdots & A^{=} & A^{+} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \cdots & A \oplus A^{-} & A^{=} & A^{+} \\ \varepsilon & \varepsilon & \varepsilon & \cdots & \varepsilon & A \oplus A^{-} & A^{=} \end{pmatrix} \text{ with}$$
$h = k_f - k_s$. Matrix $D_h$ presents an original block tri-diagonal structure: this is a square matrix, composed of a lower diagonal (square sub-matrices $A \oplus A^-$), a main diagonal (square sub-matrices $A^{=}$ ) and an upper diagonal (square sub-matrices $A^+$), with all other blocks being zero matrices ($\varepsilon$). Matrix $D_h$ is a $n.(h+1)$) x $n.(h+1)$) matrix where $n$ is the dimension of $x$.

***Theorem** 1.* (Theorem 2 in (Declerck et al. (2009))) The state trajectories of a Timed Event Graph (1) starting from $\underline{x}(k_s)$ and following the specifications defined by a P-time Event Graph (2) on horizon $[k_s, k_f]$ satisfy the following system

$$\begin{cases} X \geq D_h \otimes X \\ x(k) \geq B \otimes u(k) \text{ for } k \in [k_s+1, k_f] \\ x(k) \leq A \otimes x(k-1) \oplus B \otimes u(k) \text{ for } k \in [k_s+1, k_f] \\ x(k_s) = \underline{x}(k_s) \end{cases} \qquad (3)$$

**Remark.** It is important to note that system (3) cannot be rewritten under a fixed point form which can be analyzed by known results.

### *2.3 Relaxed system*

Equality $x(k) = A \otimes x(k-1) \oplus B \otimes u(k)$ comes from the earliest firing rule. In this part, we determine the conditions such that the determination of the trajectory only needs to use $x(k) \geq A \otimes x(k-1) \oplus B \otimes u(k)$ . Therefore, relation $x(k) \leq A \otimes x(k-1) \oplus B \otimes u(k)$ in (3) is disregarded. From system (3) which describes Problem 1, we deduce the following relaxed system

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

which presents a fixed point form.

We now characterize the sets of trajectories of systems (3) and (4).

***Property** 1.* Each trajectory of system (3) satisfies (4).

**Proof.** Immediate: As system (3) contains an additional constraint, any trajectory of this system satisfies relaxed system (4). ■

The extended space defined by (4) is now restricted by an additional condition $B \otimes u(k) = x(k)$. Contrary to the initial space (see example 1), the space is now an inf-semilattice allowing the use of the efficient algorithms of graph theory. The following theorem is the starting point of the proposed approach. As we below consider the equality $B \otimes u(k) = x(k)$ with $x(k)$ finite for $k \in [k_s+1, k_f]$, matrix $B$ is necessary row-astic.

***Theorem 2.*** (Theorem 4 in (Declerck et al. (2009))) A trajectory $X$ of (4) satisfies (3) if control $u(k)$ satisfies condition $B \otimes u(k) = x(k)$ for $k \in [k_s+1, k_f]$ .

Therefore, an (arbitrary) admissible trajectory satisfying Problem 1 can be found if we can find a trajectory satisfying the relaxed system (4) under the condition of existence of a control such that $B \otimes u(k) = x(k)$ for $k \in [k_s+1, k_f]$. We can also focus on the earliest state trajectory. Let us now determine the earliest state trajectory $X^-$ with (4). Let $E = \left( \underline{x}(k_s)^t \ \varepsilon \ \cdots \ \varepsilon \right)^t$ with $dim(E) = dim(X)$. As constraint $x(k_s) = \underline{x}(k_s)$ can be written $x(k_s) \leq \underline{x}(k_s)$ and $\underline{x}(k_s) \leq x(k_s)$, the earliest state trajectory $X^-$ is given by the resolution of $X \geq D_h \otimes X \oplus E$ with condition $\underline{x}(k_s) \geq x^-(k_s)$. The application of Kleene star by Theorem 4.75 part 1 in (Baccelli et al. (1992)) gives the lowest solution $X^- = (D_h)^* \otimes E$ with condition $\underline{x}(k_s) \geq x^-(k_s)$. Moreover, a control can be easily calculated: the greatest control to $B \otimes u(k) = x(k)$ is obviously $u(k) = B \backslash x(k)$ under the condition $B \otimes (B \backslash x(k)) = x(k)$ .

Let us now make a brief analysis of $B \otimes u(k) = x(k)$ . The analysis of this condition is out the scope of this paper and a deeper study will be proposed in a future paper. A first analysis on the condition $B \otimes u(k) = x(k)$ is given in (Declerck et al. (2009)). In the following property, the considered set of matrices includes the set of permutation matrices without being limited to this set.

***Property 2.*** If each row $i$ of the matrix $B$ at least contains a non-null element $B_{i,j}$ which is unique in the column $j$ , we can find a control $u(k)$ such that $B \otimes u(k) = x(k)$ is satisfied for any $x(k)$.

**Proof.** If $B$ is row-astic, each row $i$ contains a non-null entry $B_{ij}$ at least and a row can contain more than one element. Suppose that there is an element $B_{i,j}$ which is unique in the column $j$ . As a result of residuation is $(A \backslash b)_i = \bigwedge_{j=1}^{m} A_{ji} \backslash b_j$ where $A$ is an $m \times n$ matrix, we obtain $u_j = (B \backslash x(k))_j = B_{ij} \backslash x_i(k)$ for row $i$ and equality $B_{ij} \otimes u_j = x_i(k)$ is satisfied for any state. As the other entries $B_{ij'}$ of the row $i$ satisfy $B_{ij'} \otimes u_{j'} \leq x_i(k)$, there is a control such that $B_{i,.} \otimes u(k) = x_i(k)$ for any state. The generalisation to all rows of B is immediate. Note that different values of control vector can also satisfy condition $B \otimes u(k) = x(k)$ . ■

## 3. CONTROL WITH DESIRED OUTPUT (PROBLEM 2)

### *3.1 Objective*

In Problem 1, the objective is to calculate a state and a control trajectory of a Timed Event Graph (1) starting from $\underline{x}(k_s)$ and following specifications defined by a P-time Event Graph (2) on horizon $[k_s, k_f]$. In this part, we consider the "Just-in-time" objective and we focus on the greatest state and control trajectory where the following condition must be satisfied: $y \leq \underline{z}$ (denoted condition c)) knowing the trajectory of the desired output $\underline{z}$ on a fixed horizon $[k_s+1, k_f]$ with $h = k_f - k_s \in \mathbb{N}$. This problem is denoted Problem 2.

### *3.2 Fixed point form*

Using the previous description of the state and control trajectories (3), we rewrite the problem under a general fixed point formulation $x \leq f(x)$ which allows the resolution of Problem 2. Let $X^+ = \left( x^+(k_s)^t \ x^+(k_s+1)^t \ x^+(k_s+2)^t \ \cdots \ x^+(k_f-1)^t \ x^+(k_f)^t \right)^t$ be the greatest estimate of state trajectory $X$.

***Theorem 3.*** (Theorem 3 in (Declerck et al. (2009))) The greatest state and control trajectory of Problem 2 is the greatest solution of the following fixed point inequality system

$$\begin{cases} X \leq D_h \backslash X \\ u(k) \leq B \backslash x(k) \text{ for } k \in [k_s+1, k_f] \\ x(k) \leq [A \otimes x(k-1) \oplus B \otimes u(k)] \wedge C \backslash \underline{z}(k) \\ \text{ for } k \in [k_s+1, k_f] \\ x(k_s) \leq \underline{x}(k_s) \end{cases} \quad (5)$$

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

If condition $\underline{x}(k_s) \leq x^+(k_s)$ is satisfied, then $\underline{x}(k_s) = x^+(k_s)$ . Note that inequality (5) is equivalent to inequality (3) if we add the constraint $C \otimes x(k) \leq \underline{z}$. Therefore, the calculated state trajectory for $k \geq k_s$ is consistent with the past evolution $k \leq k_s$: In other words, the Timed Event Graph can follow calculated trajectory $X^+$ after $k_s$ which obeys the specifications defined by the P-time Event Graph.

System (5) leads to a fixed-point formulation whose general form is such that $x \leq f(x)$ where $f$ is a (min, max, +) function and can be defined by the following grammar: $f = b, x_1, x_2, \ldots, x_n \mid f \otimes a \mid f \wedge f \mid f \oplus f$ where $a, b$ are arbitrary real numbers ($a, b \in \mathbb{R}$). It is important to note that the concept of extremal solution exists in system (5) contrary to system (3).

Effective calculation of the greatest control can be made by a classical iterative algorithm of (Millan et al. (1992)) (pseudo-polynomial) which particularizes the algorithm of Kleene to (min, max, +) expressions, or more complex algorithms (Walkup (1995), Cheng et al. (2005)).

### *3.3 Relaxed system*

The following result shows that assumption $x(k) = B \otimes u(k)$ gives a simplified form $x \leq f(x)$ where $f$ is only a (min,+) function.

***Theorem 4.*** The greatest state trajectory $X^+$ of Problem 2 is the greatest solution of the following fixed point inequality system

$$\begin{cases} X \leq D_h \backslash X \\ x(k) \leq C \backslash \underline{z}(k) \text{ for } k \in [k_s+1, k_f] \\ x(k_s) \leq \underline{x}(k_s) \end{cases} \quad (6)$$

with condition $\underline{x}(k_s) \leq x^+(k_s)$ if control $u(k)$ satisfies condition $B \otimes u(k) = x(k)$ for $k \in [k_s+1, k_f]$ .

Proof

Theorem 2 shows that a trajectory $X$ of (4) satisfies (3) if control $u(k)$ satisfies condition $B \otimes u(k) = x(k)$ for $k \in [k_s+1, k_f]$ . This result holds if we add the constraint $C \otimes x(k) \leq \underline{z}$ . Therefore, we can deduce system (6) as in the previous theorem. ■

Let us calculate the greatest state of the control problem. Let

$F = \left( \underline{x}(k_s)^t \ (C\backslash \underline{z}(k_s+1))^t \ (C\backslash \underline{z}(k_s+2))^t \ \cdots \ (C\backslash \underline{z}(k_f))^t)^t \right)^t$. System of inequalities (6) becomes

$$\begin{cases} X \leq D_h\backslash X \wedge F \\ x(k) = u(k) \text{ for } k \in [k_s+1, k_f] \end{cases}$$

with condition $\underline{x}(k_s) \leq x^+(k_s)$. The application of Theorem 4.73 in (Baccelli et al. (1992)) gives the greatest solution $X^+ = D_h^* \backslash F$ with condition $\underline{x}(k_s) \leq x^+(k_s)$.

Similarly to the calculation of the earliest state trajectory, the resolution of relaxed fixed-point form (6) uses the Kleene star which improves the resolution: now, the complexity is not pseudo-polynomial but polynomial in the strongly sense.

Up to now, Problem 1 and 2 are considered on a fixed horizon. The aim of the following part is the extension of problem 2 to a control approach on a sliding horizon.

## 4. CONTROL ON A SLIDING HORIZON (PROBLEM 3): ON-LINE AND OFF-LINE ASPECTS

Let us briefly recall the technique of the predictive control on a sliding horizon. We assume that each event date of transition firing is available for current number of event k: at step $k = k_s$, $\underline{u}(k_s)$ and $\underline{x}(k_s)$ are known. A future control sequence $u(k)$ for $k \in [k_s+1, k_s+h]$ is calculated and only the first element of the optimal sequence (here $u(k_s+1)$) is applied to the process. At the next number of event $k_s+1$, the horizon is shifted: at step $k_s+1$, the problem is updated with new information $\underline{u}(k_s+1)$ and $\underline{x}(k_s+1)$ and a new calculation is performed.

However, we must guarantee the coherence of the state trajectory between each iteration, that is, the future trajectory $k \geq k_s+1$ is the extension of the past trajectory ($k \leq k_s$): more formally, the equality $\underline{x}(k_s) = x^+(k_s)$ must be satisfied otherwise the control problem 2 has no solution. If we consider that the models of the Timed Event Graph and the specifications cannot be modified, a possibility is to put the desired output back such that this problem 2 presents a solution. In this paper, we choose to solve the problem 2 with the modified desired output trajectory $z^m(k) = \underline{z}(k) \oplus y^-(k)$ for $k \in [k_s+1, k_s+h]$ where $y^-$ is an admissible trajectory. This procedure yields an optimal control for $z^m$ which can always be applied to the process. Indeed, if $u^-$and $X^-$ are the control and the state corresponding to the admissible trajectory $y^-$, then the control and the state corresponding to $z^m$ are greater.
**Remark.** The application of the control must be made after the dates of $\underline{x}(k_s)$ which are the data of the problem. As this condition must be satisfied, we have develop a technique which allows the determination of a *causal* control. This point is described in another proposed paper.

### *4.1 CPU time of the on-line control*

As control approaches using a sliding horizon need efficient algorithms, we consider the complexity of the calculation of the on-line control. We made computation tests on

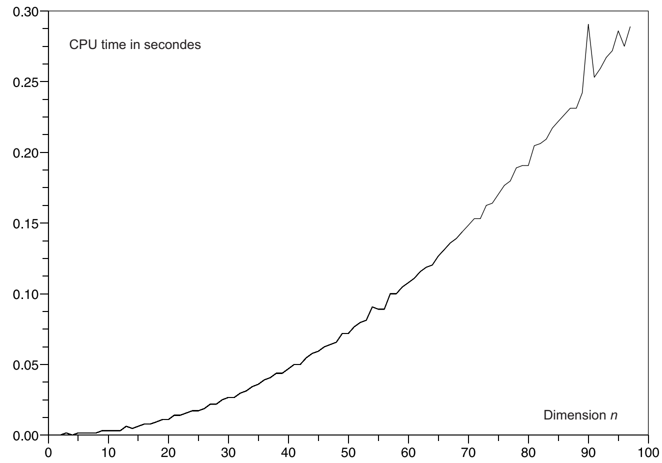


Fig. 1. On-line control: CPU times of one step for $h$=50 and randomly generated matrices until $n = 97$ transitions. The CPU times limited to 0.3 secondes only shows the efficiency of the procedure.

CPU time of the proposed approach using the max-plus toolbox in Scilab 3.1.1 with an Intel Core2 Duo 2.26 GHz. This time value includes the time of the two algorithms corresponding to the prediction procedure and the control synthesis proposed in this paper which correspond respectively to the resolution of the relaxed systems (4) and (6) of problem 1 and 2. However, the CPU time of the control does not include the calculation of the star $(D_h)^*$ which is made in the *off-line preparation* which depends on the matrices of the model and the size of the horizon. It needs the memorization of a large matrix ($(h+1).n$ x $(h+1).n$). Therefore, the calculations of the on-line control are only limited to the multiplication of matrices $(D_h)^*$ and $E$ ($X^- = (D_h)^* \otimes E$) and the left $\otimes-$residuation of $F$ by $D_h^*$ ($X^+ = D_h^* \backslash F$) in time $O(q^2)$ where $q = (h+1).n$ . In the curve of Figure 1, we have listed the CPU time needed to compute the control for horizon $h = 50$ and different dimensions $n$ of the state matrices of the event graph until $n = 97$. The matrices of the system are randomly generated ($A^= = \varepsilon$, $A^-$ and $A^+$ are full). Note that for $n = 90$, the off-line preparation which is the calculation of the kleene star with function star() of Scilab, approximately needs $2.10^3$ secondes while the on-line procedure only needs 0.28 secondes (Figure 1). Therefore, the CPU time of the on-line control is drastically reduced.

### *4.2 Example 1 continued.*

**Models.** Timed Event Graph:

$T_1 = 1.11$, $T_2 = 2$, $T_3 = 3.3$, $T_4 = 4$, $T_5 = 4.7$, $T_6 = 3$, $T_7 = 2.5$ and $T_8 = 8$.

P-Time Event Graph:

$T_{10} = 0$ , $T_{11} = 9$, $T_{12} = 1$, $T_{13} = 5$, $T_{20} = 5$ , $T_{21} = 53$, $T_{22} = 10$ and $T_{23} = 29$.

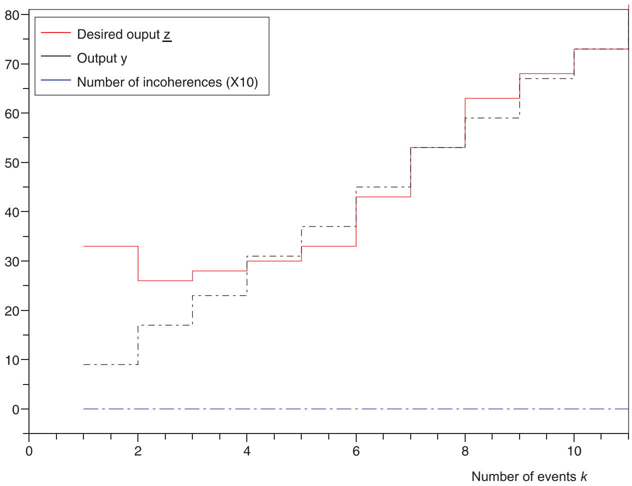


Fig. 2. One step of the simulation

**Off-line preparation.**

For $h = 2$, we have

$$D_h = \begin{pmatrix} \varepsilon & \varepsilon & -10 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & -5 & -8 & \varepsilon & \varepsilon \\ 1.11 & 5 & \varepsilon & \varepsilon & -10 & \varepsilon \\ 9 & 3.3 & \varepsilon & \varepsilon & -5 & -8 \\ \varepsilon & \varepsilon & 1.11 & 5 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 9 & 3.3 & \varepsilon & \varepsilon \end{pmatrix}.$$ The application of function star() of Scilab gives $D_h^* =$

$$\begin{pmatrix} 0 & -5 & -10 & -13 & -18 & -21 \\ 1 & 0 & -5 & -8 & -13 & -16 \\ 6 & 5 & 0 & -3 & -8 & -11 \\ 9 & 6 & 1 & 0 & -5 & -8 \\ 14 & 11 & 6 & 5 & 0 & -3 \\ 15 & 14 & 9 & 6 & 1 & 0 \end{pmatrix}.$$

**On-line control.**

Let $k_s = 1$ and $h = 10$ : the horizon is $[k_s, k_s + h] = [1, 11]$ and the problem is the determination of the control $u$ on the horizon $[2, 11]$. The initial state vector $\underline{x}(k_s)=(\,0\;\;1\,)^t$ .

The simulation is a direct application of the state equality. The analysis of Figure 2 which describes output $y$ and desired output $\underline{z}$, shows that condition a) (the process follows the specifications defined by a P-time Event Graph: the number of incoherences of the specifications is null) and condition c) (Just-in-time criteria $y \le z_m$) are satisfied.

## REFERENCES


D. Andersson and S. Vorobyov. *Fast algorithms for monotonic Discounted Linear Programs with two variables per inequality.* preprint NI06019-LAA, Isaac Newton Institute for Mathematical Sciences, Cambridge, United Kingdom, May 2006.

F. Baccelli, G. Cohen, G.J. Olsder and J.P. Quadrat. Synchronization and Linearity. An Algebra for Discrete Event Systems. available from http://maxplus.org, New York, Wiley, 1992.

P. Butkovič and G. Hegedüs. *An elimination method for finding all solutions of the system of linear equations over an extremal algebra.* Ekonomicko-Matematicky Obzor, Vol. 20, no 2, pp. 203-215, 1984.

Y. Cheng and D-Z. Zheng. *Min-max inequalities and the timing verification problem with max and linear constraints.* Discrete Event Dynamic Systems : Theory and Applications, Springer Science and Business Media, Vol 15, pp. 119–143, 2005.

E. Cohen and Nimrod Megiddo. *Improved algorithms for linear inequalities with two variables per inequality.* SIAM J. Comput., Vol 23, No 6, pp. 1313-1347, December 1994.

R.W. Cottle and A.F. Veinott. *Polyhedral sets having a least element.* Mathematical Programming 3, pp. 238-249, 1972.

P. Declerck, A. Guezzi. *Trajectory Tracking Control of a Timed Event Graph with Specifications Defined by a P-time Event Graph.* POSTA 09 Positive Systems: Theory and Applications, Valencia, Spain, 02-04 September, 2009.

P. Declerck. *From extremal trajectories to consistency in P-time Event Graphs.* available from http://www.istia.univ-angers.fr/∼declerck/, technical report, Lisa, University of Angers, September 26, 2007.

M. Gondran and M. Minoux. *Graphs and Algorithms.* John Wiley and Sons, March 1984.

A. Guezzi, P. Declerck and J.-L. Boimond. *From monotone inequalities to Model Predictive Control.* ETFA 2008, Hamburg, Germany 15-18 September 2008.

D. S. Hochbaum. *Monotonizing linear programs with up to two nonzeroes per column.* Operations Research Letters, No. 32, pp 49-58, 2004.

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

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

V. Pan. *On the complexity of a pivot step of the revised simplex algorithm.* Comp. & Maths with Appls, Vol. 11, N°11, pp.1127-1140, 1985.

M. Queyranne and F. Tardella. *Bimonotone linear inequalities and sublattices of* $\mathbb{R}^n$. Linear Algebra and its Applications 413, pp. 100-120, 2006.

G. Savard. *Introduction aux méthodes de points intérieurs.* Extrait de notes de cours, February 2001.

B. De Schutter and T. van den Boom. *Model predictive control for max-plus-linear discrete event systems.* Automatica, Vol. 37(7), pp. 1049-1056, 2001.

E. Walkup. *Optimization of linear max-plus systems with application to timing analysis* University of Washington, PhD thesis, 1995.