

**Proceedings of the 5th IFAC Workshop Available online at www.sciencedirect.com** 





IFAC-PapersOnLine 48-7 (2015) 091–096

# **Model Predictive Control for Timed Petri Nets Model Predictive Control for Timed Petri Nets Model Predictive Control for Timed Petri Nets Dimitri Lefebvre Dimitri Lefebvre Model Predictive Control for Timed Petri Nets Model Predictive Control for Timed Petri Nets**

**Havra** 75 rue Roi GREAH-University Le Havre, 75 rue Bellot, 76600 Le Havre, France

*(e-mail: dimitri.lefebvre@univ-lehavre.fr)* **Pinter** Legacy Control (e-mail: dimitri.lefebvre@univ-lehavre.fr)

contribution is to propose algorithms that incrementally compute control sequences with minimal or nearminimal duration to drive the marking of timed Petri nets (TPNs) from an initial value to a reference one. These algorithms are based on a partial exploration of the TPN reachability graph with an approach inspired from model predictive control. They include perturbation rejection and forbidden marking avoidance and from model predictive control. They include perturbation rejection and forbidden marking a volume  $\alpha$ are suitable to track trajectories in real time context when the initial and reference markings are far from each other. each other. each other. Abstract: This paper is about deadlock-free scheduling problems for discrete event systems. The main from model predictive control. They include perturbation rejection and forbidden marking avoidance and<br>are suitable to track trajectories in real time context when the initial and reference markings are far from  $f(x)$  is each other. They include perturbation regeneration regeneration regeneration regeneration  $\mathcal{L}$ each other. Suitable trajectories in real time context when the initial and reference markings are far from the initial and reference matches are from the initial and reference markings are from the initial and reference m

© 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.  $\cup$  2015, IF F  $\odot$  2015 IF  $t$ 

### 1. INTRODUCTION 1. INTRODUCTION 1. INTRODUCTION 1. INTRODUCTION

Petri nets (PNs) are useful for the scheduling and supervisory control problems with discrete event systems (DESs) (Cassandras, 1993) because they combine, in a comprehensive way, intuitive graphical representations and powerful analytic expressions. In particular timed PNs are used when temporal specifications are concerned (David and Alla, 1992). Pioneer works about the scheduling of manufacturing systems have been developed with an adaptation of the A\* algorithm to the PN structures (Lee and DiCesare, 1994). The method partially explores the reachability graph according to a heuristic function (usually based on the makespan) and the performance basically lies in how good this function is. Numerous<br>improvements have been further developed including province improvements have been further developed including pruning of non-promising branches (Reyes-Moro et al., 2002), backtracking limitation (Xiong and Zhou, 1998), determination of lower bounds for the makespan (Jeng and Chen, 1998), or use of structural analysis (Abdallah et al. 2002). In order to limit the use of resources and jobs and also to maintain some<br>heliotricianal properties on liveness or controllability. (Didoden behavioral properties as liveness or controllability (Didedan and Alla, 2008) supervisory control methods have been also developed (Ramadge and Wonham, 1987; Basile et al., 2013). The principle of supervisory control is to avoid some forbidden markings by adding generalized mutual exclusion constraints over an initial unconstrained PN model. Constraints are generally implemented with monitor places. Only a few results combine scheduling and supervisory control in the same approach. Search in the partial reachability graph (Abdallah et al. 2002), genetic algorithms (Xing et al., 2012) and heuristic functions based on the firing vector (Lei et al., 2014) have been used to find non-optimal deadlock-free schedules. control problems with discrete event systems (DESs) (Cassandras, 1993) because they combine, in a comprehensive way, intuitive graphical representations and powerful analytic<br>way, intuitive graphical representations and powerful analytic way, intuitive graphical representations and powerful analytic expressions. In particular timed Privs are used when temporal<br>expressions are concerned (David and Alla, 1992). Diancer specifications are concerned (David and Alla, 1992). Pioneer been developed with an adaptation of the A\* algorithm to the<br>DN structures (Lee and DiCesses, 1004). The mathed neutrally  $P_{\text{N}}$  structures (Lee and DiCesare, 1994). The method partially explores the reachability graph according to a heuristic<br>function (usually head on the makespan) and the nonformance  $f(x)$  function (excellent on the material on the material on the performance basically lies in how good this function is. Numerous basically lies in how good this function is. Numerous vasically lies in how good this function is. Numerous of non-promising branches (Reyes-Moro et al., 2002), backof non-promising branches (Keyes-Moro et al., 2002), back-<br>tracking limitation (Xiong and Zhou, 1999), determination of Exercise Initiation (2001) and 2100, 1990), determination of<br>lower bounds for the makespan (Jeng and Chen, 1998), or use of structural analysis (Abdallah et al. 2002). In order to limit the use of resources and jobs and also to maintain some the use of resources and jobs and also to maintain some behavioral properties as liveness or controllability (Didedan behavioral properties as liveness or controllability (Didedah<br>and Alla, 2008) supervisory control methods have been also developed (Ramadge and Wonham, 1987; Basile et al., 2013). and Alla, 2008) supervisory control methods have been also The principle of supervisory control is to avoid some forbidden<br>mortings by adding concreting mutual avaluation constraints marking by adding generalized mutual exclusion constraints markings by adding generalized mutual exclusion constraints are an initial are against mutual Constraints are generally implemented with monitor places. Only a few results<br>generally implemented with monitor places. Only a few results generally impremented with moment places. Only a few results combine scheduling and supervisory control in the same approach. Search in the partial reachability graph (Abdahan et<br>al. 2002), genetic algorithms (Xing et al., 2012) and heuristic  $f(x) = f(x)$  functions dependently (Allignoise and  $f(x) = f(x) + 1$ , 2014) have been runctions based on the firing vector (Lei et al., 2014) have been improvements have been further developed including pruning of non-promising branches (Reyes-Moro et al., 2002), back-Petri nets (PNs) are useful for the scheduling and supervisory (Cassandras,  $1993$ ) because they combine, in a comprehensive  $\epsilon$  expressions. In particular time  $\epsilon$  PNs are used when temporal specifications are concerned (David and Alia, 1992). Ploneer works about the scheduling of manufacturing systems have PN structures (Lee and DICesare,  $1994$ ). The method partially Improvements nave been further developed including pruning of non-promising branches (Reyes-Moro et al.,  $2002$ ), backracking ilmitation ( $\lambda$ long and  $\lambda$ nou, 1998), determination of developed (Kamadge and Wonnam,  $1987$ , Basile et al.,  $2013$ ). markings by adding generalized mutual exclusion constraints over an initial unconstrained PN model. Constraints are combine scheduling and supervisory control in the same approach. Search in the partial reachability graph (Abdallah et  $\frac{1}{2000}$ functions based on the firing vector (Left et al.,  $2014$ ) used to find non-optimal deadlock-free schedules.

A common limitation of the scheduling and supervisory control methods is that they require a large computational effort so that they are generally implemented off-line and are not suitable for real time scheduling or other on-line control applications. Real time scheduling of other on-line control<br>applications. Real time control has been better investigated<br>with continuous PNs (Silva and Bosalda, 2004). In particular with continuous PNs (Silva and Recalde, 2004). In particular, model predictive control (MPC) has been successfully used with continuous timed PNs (Mahulea et al., 2008) to optimize a cost function that measures the distance with reference marking. But the limitation of continuous PNs and other continuous time approaches (Apaydin-Özkan et al., 2011) is used to find non-optimal deadlock-free schedules. common minimum of the senegating and supervisory como memors is that they require a large computational not suitable for real time scheduling or other on-line control not suitable for real time scheduling or other on-line control<br>applications. Real time control has been better investigated applications. Real three control has been better investigated<br>with continuous PNs (Silva and Recalde, 2004). In particular, with continuous PNs (Silva and Recalde, 2004). In particular,  $mg/dt$  we define  $m$  and predictive control (MPC) has been successfully used  $m$  the optimize model  $m$ a cost function that measures the distance with reference<br>measures the distance with reference a cost function that measures the urstance with reference marking. But the ilmitation of continuous PNs and other extraction in the selections and supervisory<br>control methods is that they require a large computational<br>effort so that they are generally implemented off-line and are with continuous PNs (Silva and Recalde, 2004). In particular, A common imitation of the scheduling and supervisory control methods is that they require a large computational model predictive control (MPC) has been successfully used with continuous time  $\alpha$  PNs (Manuea et al., 2008) to optimize

that a continuous abstraction of DESs is made at first. that a continuous abstraction of DESs is made at first. that a continuous abstraction of DESs is made at first. Consequently, the continuous control strategies cannot be directly implemented with DESs and the properties of the closed-loop continuous models are not preserved with the discrete systems. Consequently, the continuous control strategies cannot be<br>directly implemented with DESs and the proportion of the Consequently, the commutation control strategies cannot be encely implemented with DESs and the properties of the discrete systems. The main contribution of this work is to propose a method for closed-loop continuous models are not preserved with the discrete systems. discrete systems. The main contribution of this work is to propose a method for  $\alpha$  continuous abstraction of DESS is made at first. directly implemented with DESS and the properties of the

The main contribution of this work is to propose a method for timed Petri nets that incrementally computes control sequences with minimal or near-minimal duration in order to reach a reference state from an initial one. The method, is inspired from MPC approach: at each step it evaluates and optimizes a performance criterion over a prediction horizon and then apply the first action of the computed control sequence. Consequently the method is robust to perturbations induced by the firing of uncontrollable transitions. It works directly with the discrete model and continuous abstractions are no longer required. As scheduling methods, it is based on a partial exploration of the PN reachability graph but limits this exploration to a narrow neighborhood of the current marking and updates on-line the control actions when uncontrollable transitions fire. The approach is suitable for bounded or unbounded timed PNs with weighted arcs. As supervisory control methods, it avoids forbidden markings but does not encode the constraints in the PN structure so that constraints can immediately be added, removed or modified. The paper is organized as follows. In Section 2, timed Petri nets systems and control problems are introduced. Section 3 proposes Algorithms that compute on-line near-minimal time firing sequences for real-time control applications. The approach includes perturbation rejection and forbidden marking avoidance. timed Petri nets that incrementally computes control requences with minimal of hear-minimal quiation in order to inspired from MPC approach: at each step it evaluates and inspired from MPC approach: at each step it evaluates and optimizes a performance criterion over a prediction nonzon<br>and then apply the first action of the computed control and their apply the mist action of the computed control induced by the firing of uncontrollable transitions. It works directly with the discrete model and continuous abstractions are no longer required. As scheduling methods, it is based on  $\alpha$ are no longer required. As scheduling methods, it is based on exploration of the EIN reachability graph out finites this<br>exploration to a narrow neighborhood of the current marking<br>and undeter on line the central ections when uncentrallely capionation to a narrow neighborhood or the current marking transitions fire. The approach is suitable for bounded or unbounded timed  $\sum_{n=1}^{\infty}$   $\sum_{n=1}^{\infty}$  with weighted arcs. As supervisory example the constraints in the DN structure as that constraints encode the constraints in the PN structure so that constraints encode the constraints in the PN structure so that constraints organized as follows. In Section 2, timed Petri nets systems and control problems are introduced. Section 3 proposes Algorithms that compute on-line near-minimal time fining exigorumns that compute on-the hear-imminar time ining<br>sequences for real-time control applications. The approach sequences for real-time control applications. The approach  $\frac{1}{2}$ The main contribution of this work is to propose a method for directly with the discrete model and continuous abstractions includes perturbation rejection and forbidden marking avoidance. avoidance. the main contribution of this work is to propose a method for umed retri nets that incrementally computes control reach a reference state from an initial one. The method, is sequence. Consequently the method is robust to perturbations. are no longer required. As scheduling methods, it is based on a partial exploration of the PN reachability graph but limits this and updates on-line the control actions when uncontrollable unbounded timed PINS with weighted arcs. As supervisory organized as ionows. In Section 2, timed Petri nets systems Algoriums that compute on line near-minimal time firing

### 2. CONTROL DESIGN FOR TIMED PNs 2. CONTROL DESIGN FOR TIMED THIS 2. CONTROL DESIGN FOR TIMED PNs

A PN structure is defined as  $G = \langle P, T, W_{PR}, W_{PQ} \rangle$ , where  $P =$  ${P_1, \ldots, P_n}$  is a set of *n* places and  $T = {T_1, \ldots, T_q}$  is a set of *q* transitions of labels  $\{1,...,q\}$ ,  $W_{PO} \in (\mathbb{N})^{n \times q}$  and  $W_{PR} \in (\mathbb{N})^{n \times q}$ are the post and pre incidence matrices (N is the set of non-<br>negative integer numbers) and  $W-W$ ,  $W$  is the incidence negative integer numbers), and  $W = W_{PO} - W_{PR}$  is the incidence<br>matrix  $\leq C M \geq$  is a DN system with initial marking M and matrix.  $\langle G, M_1 \rangle$  is a PN system with initial marking  $M_1$  and  $M_2$ . (N) separate the DN system with initial marking  $M_1$  and  $M \in (\mathbb{N})^n$  represents the PN marking vector. A transition  $T_j$  is enabled at marking *M* if and only if (iff)  $M \ge W_{PR}(:, j)$ , where  $W_{PR}(:, j)$  is the column *j* of pre incidence matrix; this is denoted as  $M[T_j] > M$  leads *T<sub>j</sub>* is enabled, it may fire, and when  $T_j$  fires  $P(X|Y)$  is a set of *n* places and  $T = (T - T)$  is a set of *q*<sup>{</sup>  $\{F_1, \ldots, F_n\}$  is a set of *n* places and  $\mathbf{I} = \{I_1, \ldots, I_q\}$  is a set of  $q$ <br>transitions of labels  $(1, \ldots)$   $W_r = (N!)^{n \times q}$  and  $W_r = (N!)^{n \times q}$ dialisticulus of factors  $\{1,...,q\}$ ,  $WPO \in \Gamma(Y)$  and  $WPR \in \Gamma(Y)$  $M = (N/M)$  represents the PN marking vector. A transition *T* is  $M \in (N)$  represents the FN marking vector. A transition  $T_j$  is<br>enabled at marking M if and only if (iff)  $M \ge W_{PR}(:, j)$ , where chaviculate individual *W* in and only if  $(III)$   $M \leq WPR(t, y)$ , where *M PR(., j)* is the column *j* of pre-incidence matrix, this is denoted  $M[T] > W$  hen *T* is enabled, it may fire and when *T* fires A PN structure is defined as  $G = \mathcal{F}$ , *P*, WPR, WPO<sup>2</sup>, where **P**<br>(*D*, *D*) is a set of *p* places and  $T = (T_x - T)$  is a set of *g*  ${P_1, ..., P_n}$  is a set of *n* places and  ${T_1, ..., T_q}$  is a set of *q*<br> ${P_2, ..., P_n}$  is a set of *Place is*  ${H_1, ..., H_q}$  is a set of *q* transitions of labels  $\{1,...,q\}$ ,  $WPO \in (N)^{n-q}$  and  $WPR \in (N)^{n-q}$ <br>are the post and pro-incidence metrics. (N is the set of pop are the post and pre-incidence matrices ( $N$  is the set of non-<br>positive integer numbers) and  $W = W$  is the incidence as *M*  $\left[ T_j \right]$   $\sim$ . When  $T_j$  is enabled, it may fire, and when  $T_j$  fires A PN structure is defined as *G* = <*P, T, WPR, WPO*>, where *P* =  ${P}$ , *P* is a set of *n* place and *T* = {*P<sub>1</sub>*,  ${P}$ ,  ${P}$ ,  ${P}$ ,  ${P}$ ,  ${P}$ ,  ${P}$  is a set of *n*  ${P}$  $\{P_1, \ldots, P_n\}$  is a set of *n* places and  $I = \{I_1, \ldots, I_q\}$  is a set of *q* transitions of labels  $\{1,...,q\}$ ,  $W_{PO} \in \{N\}$  and  $W_{PR} \in \{N\}$  and are the post and pre-incidence matrices (N is the set of nonmatrix matrix is the increase of  $M = W_{PO} - W_{PR}$  is the increase of  $G_{M}$  and  $W = W_{PO} - W_{PR}$  is the increase of  $G_{M}$  $M \in (N)$  represents the PN marking vector. A transition  $I_j$  is  $W_{PR}(\cdot, j)$  is the column *J* of pre includence matrix; this is denoted

2405-8963 © 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved. Peer review under responsibility of International Federation of Automatic Control. **10.1016/j.ifacol.2015.06.478** 

once, the marking varies according to  $\Delta M = M' - M = W(:, i)$ . This is denoted as  $M[T_i > M']$ . With infinite server semantic, any transition may fire several times simultaneously and the maximal number of simultaneous firings for transition  $T_i$  at marking *M* is given by  $n_i(M)$ :

$$
n_j(M) = \min\{\lfloor m_k / w^{PR} \rfloor : P_k \in {}^{\circ}T_j\} \tag{1}
$$

where  $\mathbf{r}_j$  stands for the set of  $T_j$  upstream places,  $m_k$  is the marking of place  $P_k$ ,  $w^{PR}$ <sub>kj</sub> is the entry of matrix  $W_{PR}$  in row  $k$ and column *j*. A marking *M* is said to be reachable from initial marking  $M_I$  if there exists a firing sequence  $\sigma$  such that (s.t.)  $M_I$  [ $\sigma$  >*M* and *R*(*G, M<sub>I</sub>*) is the set of all reachable markings from initial marking *MI*.

Timed Petri nets are PNs whose behaviors are constraint by temporal specifications. For this reason, timed PNs have been intensively used to describe DESs like production systems. This paper concerns T-timed PNs (TPNs) (Ramchandani, 1973) under infinite server semantic and preselection policy. A TPN system is defined as  $\leq G$ ,  $M_I$ ,  $D_{min}$  where  $D_{min} = (d_{min})$  $j \in (\mathbf{R}^+)^q$  is the vector of the minimal firing delays of the transitions  $(\mathbb{R}^+$  the set of non-negative real numbers). Firing delays are given in time units (TUs). Before firing transition  $T_i$ , tokens are reserved in the  $T_j$  preset places during a duration  $d_j \geq d_{minj}$ . A timed firing sequence  $\sigma$  of length  $|\sigma| = h$  fired at marking  $M_I$  is defined as:

$$
\sigma = T(j_1, t_1)T(j_2, t_2)\dots T(j_h, t_h)
$$
\n
$$
(2)
$$

where  $j_1, \ldots, j_h$  are the labels of the transitions and  $t_1, \ldots, t_h$ represent the dates of the firings that satisfy  $0 \le t_1 \le t_2 \le ... \le t_h$ . The sequence (2) leads to the timed marking trajectory (3):

$$
TR(\sigma) = M_I \left[ T(j_I, t_I) > M(1) \dots \left[ T(j_h, t_h) > M_F \right] \right] \tag{3}
$$

where  $M(1),...,M(h-1)$  are the intermediate visited markings and  $M_F$  is the final marking. The time criteria  $J_T$  is defined according to the duration of  $\sigma$  or equivalently the duration of  $TR(\sigma)$ :

$$
J_T(\sigma) = J_T(TR(\sigma)) = t_h \tag{4}
$$

For control issues, the marking vector is assumed to be observable and the set of transitions *T* is divided into 2 disjoint subsets *T<sub>C</sub>* and *T<sub>NC</sub>* s.t.  $T = T_C \cup T_{NC}$ . *T<sub>C</sub>* is the subset of controllable transitions and  $T_{NC}$  the subset of uncontrollable ones. The firing of controllable transitions that are enabled and that satisfy the temporal specifications of TPNs can be enforced or avoided by the controller whereas the firing of uncontrollable transitions occurs according to events that are not driven by the controller (for example internal events of the modeled system). When a conflict exists between controllable and uncontrollable transitions, priority is always given to the uncontrollable ones that fire first. Conflicts between controllable transitions are solved by the controller, whereas conflicts between uncontrollable transitions are solved by a policy that is unknown for the controller. A firing sequence of controllable transitions is called a control sequence.  $\mathbf{R}/G_C$ ,  $M_I$ )  $\subseteq$  *R*(*G, M<sub>I</sub>*) is the set of all reachable markings from *M<sub>I</sub>* by firing only controllable sequences where  $G_C$  is the TPN structure obtained by removing all uncontrollable transitions. Depending on the control application, some forbidden markings (i.e. to be avoided by the marking trajectories) may

be also specified. For this purpose, the function *LEGAL* is defined for any marking  $M \in \mathbf{R}(G_C, M)$  as  $LEGAL(M) = 0$  if the marking is forbidden else *LEGAL(M)* = 1 and  $\mathbf{R}_L(G_C, M_I)$  $\subseteq$  *R*(*Gc, M<sub>I</sub>*) is the set of markings that are legal (i.e.  $LEGAL(M) = 1$ ).

The objective of the proposed control design is to reach a reference marking  $M_{ref} \in \mathbf{R}_L(G_C, M_I)$  starting from initial marking  $M<sub>I</sub>$  with a control sequence of minimal or nearminimal duration that visits no forbidden marking.

# 3. COMPUTATION OF NEAR-MINIMAL TIME SEQUENCES

## *3.1 Minimal time control sequences*

Let us consider  $M_{ref} \in \mathbf{R}_L(G_C, M_I)$  and define  $\sum^* (M_L M_{ref})$  as the set of control sequences of minimal time *tmin\** that drive the marking vector of a TPN from *MI* to *Mref* by avoiding forbidden markings. For any integer  $H \geq 0$  and any marking  $M \in \mathbf{R}_L(G_C)$ *MI)*, the following notations are also introduced:

- $\sum (M,H)$  is the set of control sequences  $\sigma$  issued from *M* s.t. *TR(* $\sigma$ ) visits no forbidden marking and  $|\sigma| \leq H$ .
- $\sum (M, M_{ref}, H) \subseteq \sum (M, H)$  is the subset of sequences that drive the marking vector to *Mref*.
- $\sum^* (M, M_{ref}, H) \subseteq \sum (M, M_{ref}, H)$  is the subset of sequences of minimal duration *tmin* according to the temporal specifications given by *Dmin*.

Standard algorithms based on A\* or D\* search methods (Lee and DiCesare, 1994) have been developed to compute control sequences for untimed Petri nets (Lefebvre and Leclercq 2014; 2015) and can be easily extended to compute  $\mathcal{L}^*(M_L M_{ref}, H)$ . The main difficulty of such algorithms is the exponential complexity with respect to (wrt) the depth *H* of the exploration. If numerous firings are need for reaching the reference marking, a large depth is required to ensure convergence. As a consequence, standard algorithms are time consuming and definitively not acceptable for on-line use or when the initial and reference markings are far from each other.

## *3.2 Double local optimization algorithm*

To limit the computational complexity of the standard algorithms, an approach inspired from the MPC is proposed in this paper. Such approach anticipates the evolution of the system in order to achieve the control objective. At each step, it minimizes, eventually under some constraints, a performance criterion over a future horizon. The search depth *H* is the prediction horizon. A sequence of control actions is obtained and only the first action of this sequence is applied (Richalet et al., 1978). The prediction starts again from the new state reached by the system. Due to the parallel use of prediction and optimization processes, MPC has been proved to be robust to perturbations and leads to efficient controllers used in many industrial applications (Camacho and Bordons, 2007). In the case of PNs, MPC has been investigated for the control design of continuous timed PN systems (Mahulea et al., 2008), for untimed discrete PNs (Lefebvre and Leclercq 2014; 2015) and for a particular class of DESs that are linear in (Max-Plus) algebra (De Schutter and Van den Boom, 2001). It this paper the method developed in (Lefebvre and Leclercq 2014; 2015) is extended for TPNs.

Download English Version:

# <https://daneshyari.com/en/article/709094>

Download Persian Version:

<https://daneshyari.com/article/709094>

[Daneshyari.com](https://daneshyari.com)