Available online at www.sciencedirect.com
ScienceDirect
IFAC-PapersOnLine 48-7 (2015) 091–096

Model Predictive Control for Timed Petri Nets

Dimitri Lefebvre
GREAH–University Le Havre, 75 rue Bellot, 76600 Le Havre, France
(e-mail: dimitri.lefebvre@univ-lehavre.fr)

Abstract: This paper is about deadlock-free scheduling problems for discrete event systems. The main contribution is to propose algorithms that incrementally compute control sequences with minimal or near-minimal 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 are suitable to track trajectories in real time context when the initial and reference markings are far from each other.

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 improvements have been further developed including pruning of non-promising branches (Reyes-Moro et al., 2002), back-tracking 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 behavioral properties as liveness or controllability (Dideban 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.

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 control has been better investigated 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 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.

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.

2. CONTROL DESIGN FOR TIMED PNs

A PN structure is defined as $G = \langle \boldsymbol{P}, \boldsymbol{T}, W_{PR}, W_{PO} \rangle$, where $\boldsymbol{P} = \{P_1,\ldots, P_n\}$ is a set of n places and $\boldsymbol{T} = \{T_1,\ldots, T_q\}$ is a set of q transitions of labels $\{1,\ldots,q\}$, $W_{PO} \in (\mathbf{N})^{n\times q}$ and $W_{PR} \in (\mathbf{N})^{n\times q}$ are the post and pre incidence matrices (**N** is the set of non-negative integer numbers), and $W = W_{PO} - W_{PR}$ is the incidence matrix. $\langle G, M_I \rangle$ is a PN system with initial marking M_I and $M \in (\mathbf{N})^n$ represents the PN marking vector. A transition T_j is enabled at marking M if and only if (iff) $M \geq W_{PR}(:, j)$, where $W_{PR}(:, j)$ is the column j of pre incidence matrix; this is denoted as $M\,[T_j >$. When T_j is enabled, it may fire, and when T_j fires

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 $\Delta M = M' - M = W(:, j)$. This is denoted as $M [T_j > M'$. With infinite server semantic, any transition may fire several times simultaneously and the maximal number of simultaneous firings for transition T_j at marking M is given by $n_j(M)$:

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

where ${}^\circ\boldsymbol{T_j}$ stands for the set of T_j upstream places, m_k is the marking of place P_k, w^{PR}_{kj} 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 σ such that (s.t.) $M_I [\sigma > M$ and $\boldsymbol{R}(G, M_I)$ is the set of all reachable markings from initial marking M_I.

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 $<G, M_I, D_{min}>$ where $D_{min} = (d_{min\ j}) \in (\mathbf{R}^+)^q$ is the vector of the minimal firing delays of the transitions ($\mathbf{R}^+$ the set of non-negative real numbers). Firing delays are given in time units (TUs). Before firing transition T_j, tokens are reserved in the T_j preset places during a duration $d_j \geq d_{min\ j}$. A timed firing sequence σ of length $|\sigma| = h$ fired at marking M_I is defined as:

$$\sigma = T(j_1, t_1)T(j_2, t_2)\ldots T(j_h, t_h) \quad (2)$$

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 $0 \leq t_1 \leq t_2 \leq \ldots \leq t_h$. The sequence (2) leads to the timed marking trajectory (3):

$$TR(\sigma) = M_I [T(j_1, t_1) > M(1)\ldots[T(j_h, t_h) > M_F. \quad (3)$$

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

$$J_T(\sigma) = J_T(TR(\sigma)) = t_h \quad (4)$$

For control issues, the marking vector is assumed to be observable and the set of transitions $\boldsymbol{T}$ is divided into 2 disjoint subsets $\boldsymbol{T_C}$ and $\boldsymbol{T_{NC}}$ s.t. $\boldsymbol{T} = \boldsymbol{T_C} \cup \boldsymbol{T_{NC}}$. $\boldsymbol{T_C}$ is the subset of controllable transitions and $\boldsymbol{T_{NC}}$ 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. $\boldsymbol{R}(G_C, M_I) \subseteq \boldsymbol{R}(G, M_I)$ is the set of all reachable markings from M_I 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 $M \in \boldsymbol{R}(G_C, M_I)$ as $LEGAL(M) = 0$ if the marking is forbidden else $LEGAL(M) = 1$ and $\boldsymbol{R_L}(G_C, M_I) \subseteq \boldsymbol{R}(G_C, M_I)$ 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 $M_{ref} \in \boldsymbol{R_L}(G_C, M_I)$ starting from initial marking M_I with a control sequence of minimal or near-minimal duration that visits no forbidden marking.

3. COMPUTATION OF NEAR-MINIMAL TIME SEQUENCES

3.1 Minimal time control sequences

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

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

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 $\Sigma^*(M_I,M_{ref},H)$. 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.