International Journal of Automation and Computing 15(1), February 2018, 25-38
DOI: 10.1007/s11633-016-1046-7

Model Predictive Control for Discrete and Continuous Timed Petri Nets

Marwa Taleb Edouard Leclercq Dimitri Lefebvre
Laboratory GREAH, University of Le Havre, Le Havre, France

Abstract: The goal of this paper is to propose a unique control method that permits the evolution of both timed continuous Petri net (TCPN) and T-timed discrete Petri net (T-TDPN) from an initial state to a desired one. Model predictive control (MPC) is a robust control scheme against perturbation and a consistent real-time constraints method. Hence, the proposed approach is studied using the MPC. However, the computational complexity may prevent the use of the MPC for large systems and for large prediction horizons. Then, the proposed approach provides some new techniques in order to reduce the high computational complexity; among them one is taking constant control actions during the prediction.

Keywords: Model predictive control, timed continuous Petri net (TCPN), T-timed discrete Petri net (T-TDPN), fluidification, optimization, constant control.

1 Introduction

Petri net (PN) is a mathematical formalism with a graphical paradigm successfully used for modeling, analysis and control of discrete event systems (DES) such as manufacturing processes[1−6], telecommunications[7], logistic and traffic systems[8, 9]. When timed discrete Petri nets (TDPN) are considered, numerous contributions based on scheduling control have been developed. The control sequence is usually determined thanks to the use of heuristic functions based on Dijkstra and A* algorithms which just generate the required portion of the reachability graph that contains the optimal path from the initial to the final marking[10]. A hybrid enhanced version of these previous methods called dynamic look-ahead stage search whose purpose is to reduce the search space is presented in [3]. In the same context, and in order to minimize the maximum completion time, reactive graph search algorithm based on real time A* and rule-based supervision have been applied to TDPN[11]. Another on-line method is based on several heuristics that reduce the backtracking and discard unnecessary schedules in order to find the feasible schedules of real-time systems[12]. Local suboptimal search methods in reachability graph have been also proposed in [13]. Unfortunately, the computational effort required to explore the reachability graph limits the use of most of the previous methods when real time control applications are considered. In particular, the methods are not applicable for systems with a large number of states. This problem is known as the state explosion problem. Various studies over the past 30 years have been carried out in order to deal with this problem. It has been shown that an efficient solution to tackle the state explosion problem consists of "fluidification" (or "continuization") which represents a usual technique introduced in [14, 15], to relax the original integrity constraints[16, 17]. The motivations, validity, advantages and limitations of the fluidication have been discussed in detail in [18] (we invite the reader to consider this review paper for a nice discussion). When time is associated with the firing of the transitions, the resulting net is called timed continuous PN (TCPN). The latter constitutes a continuous approximation of the DES[19]. Many server approximations exist in the literature. In this paper, we consider TCPN under infinite server semantics. This latter can be submitted to external control actions in order to control the system and to reach a desired state. The resulting PN is called controlled TCPN system. The crucial question of the control design is how to compute the control actions. During the last years, various methods have been developed to design the optimal control under some constraints. As far as the TCPN is concerned, control structure, which consists of a linear programming problem, is developed and solved on-line. The authors in [20] have studied the optimal steady-state control, while in [21], the goal was to minimize the time of the trajectory between the initial state to the target one using linear and bi-linear programming. However, in [22], the desired stationary marking vector and the desired asymptotic firing rate vector have been reached thanks to the feedback control strategy. In [23], piecewise-constant control actions are proposed according to the regions defined from the synchronizations of the net. In [24], the authors introduce gradient-based controllers to minimize the quadratic instantaneous error between the measured output and the desired one.

Research Article
Manuscript received December 8, 2015; accepted June 17, 2016; published online February 21, 2017
This work was supported by the region Haute-Normandie Project (Nos. CPER-SER-DDSMRI 2013, 2014 and CPER-SER-SEL 2015).
Recommended by Associate Editor Hong-Nian Yu

Another very attractive strategy is called model predictive control (MPC). This method is robust against perturbation and it represents a consistent real-time constraints method. For these reasons it is usually and widely used in industry. Its basic idea consists of minimizing, at each time step, a performance criterion over a future horizon, a sequence of control actions is then computed and only the first action is applied to the system[25]. In [16, 26, 27], MPC-based algorithms have been proposed for TCPN. But complexity issues prevent the use of the MPC approach for large prediction horizons.

In this research work, we consider the MPC-based control design of TCPN and T-TDPN. The control of DES results from the fluidification of the T-TDPN that avoids the exploration of the state graph of the net[28]. Our contributions aim to reduce the computational complexity by using constant control actions during the whole prediction horizon, to introduce terminal constraints in the neighborhood of the reference marking in order to ensure the stability of the closed-loop system. In summary, this paper develops an implicit approach called model predictive constant control (MPCC) under an adaptive prediction horizon. This contribution is a sequel of our previous studies in [29, 30].

The article is organized as follows: Section 2 presents the basic concepts about discrete and continuous PN systems and introduces some notations to be used. In Section 3, the fundamental idea of MPCC is presented. Section 4 deals with the application of MPCC to control T-TDPN. Finally, conclusions and some prospects are drawn in Section 5.

2 Petri nets

The reader is assumed to be familiar with continuous and discrete Petri nets (for an introduction, we invite the reader to consider[31−33]).

Let us, first, introduce the following notations:

1) M_0: the initial marking.
2) M: the marking in the autonomous PN system.
3) $t^d \in T$: a discrete transition in the T-TDPN.
4) $M^d \in \mathbf{N}^n$: a discrete marking in the T-TDPN.
5) $t^c \in T$: a continuous transition in the TCPN.
6) $M^c \in \mathbf{R}^n_{\geq 0}$: a continuous marking in the TCPN.
7) $D_{\min}$: the minimal duration of transitions firing.
8) λ: the maximum firing speed.
9) θ: the sampling period.
10) Z: the number of regions.
11) A_z: the constraint matrix of a region R_z.
12) N: the prediction horizon of the MPCC.
13) M_{ref}: the desired marking.
14) X_{ref}: the desired flow.
15) (M_{ref}, X_{ref}): the desired configuration.
16) ϵ: a small strictly positive parameter that defines a neighborhood of the desired configuration.
17) $t^{d\_opt} \in T$: the optimal discrete transition in the T-TDPN.
18) σ: an untimed control sequence.
19) σ^{timed}: a timed control sequence.
20) H: the length of the control sequence.

2.1 Untimed Petri net systems

A PN system is a couple $\langle Net, M_0\rangle$. The net structure Net is represented by the quadruplet: $\langle P, T, W_{pr}, W_{po}\rangle$ where $P = \{p_i\}_{i=1,\cdots,n}$ and $T = \{t_j\}_{j=1,\cdots,q}$ are two nonempty finite sets composed respectively of $n = |P|$ places and $q = |T|$ transitions. The matrices W_{pr} and $W_{po} \in \mathbf{N}^{n\times q}_{\geq 0}$ are respectively the input incidence and the output incidence matrices which define the static structure of the net: $\forall (p_i, t_j) \in P \times T$, $W_{pr}(p_i, t_j)$ and $W_{po}(p_i, t_j)$ represent respectively the weights of the arcs from p_i to t_j and from t_j to p_i. Let $W = W_{po} - W_{pr}$ denote the incidence matrix. For each place p_i, the current marking is defined by $m_i \in \mathbf{N}_{\geq 0}$. The marking vector is denoted as $M \in \mathbf{N}^n_{\geq 0}$. $M_0 \in \mathbf{N}^n_{\geq 0}$ denotes the initial marking of the PN. Each node $v \in P \cup T$ has its sets of input and output nodes respectively denoted as $^\circ v$ and v°. Let $\eta(t_j, M)$ be the enabling degree of a transition t_j. This latter is given by (1):

$$\eta(t_j, M) = \min_{p_i \in {}^\circ t_j} \left\{ \left\lfloor \frac{m_i}{W_{pr}(p_i, t_j)} \right\rfloor \right\} \tag{1}$$

where $\lfloor \cdot \rfloor$ is the integer part of $(\cdot)$. A transition t_j is enabled at M if and only if (iff) $\eta(t_j, M) > 0$. In this case, $\eta(t_j, M)$ measures the maximum number (or quantity) of firings of the enabled transition t_j. Hence, the amount in which t_j can fire is restricted to an integer $\alpha \in \mathbf{N}_{\geq 0}$ such that $0 < \alpha \leq \eta(t_j, M)$. The marking variation ΔM resulting from the firing of t_j is defined by (2):

$$\Delta M = \alpha \times W(:, t_j) \tag{2}$$

where $W(:, t_j)$ is the j-th column of the incidence matrix W. The series of transitions which successively fire from an initial state M_0 to another one M is represented by the firing sequence σ. The mathematical representation is given by $M_0 [\sigma \rightarrow M]$. Such a sequence is characterized by the firing count vector $\Gamma = (\gamma_j)_{j=1,\cdots,q}$ where γ_j stands for the cumulative number of firings of t_j in the sequence σ. The PN reachability set $R(Net, M_0)$ denotes the set of all markings which are reachable from the initial marking through a finite sequence σ. Thus, if $M \in R(Net, M_0)$, the state equation is written as (3):

$$M = M_0 + W \times \Gamma. \tag{3}$$

Certain useful invariant laws can be found thanks to the right and left natural annuler of the incidence matrix. If $\exists Z \geq 0$ such that $Z^t \times W = 0$, Z is called a P-semi-flow. If $Z > 0$, the PN is said to be consistent. Similarly, if $\exists Y \geq 0$ such that $W \times Y = 0$, Y is called a T-semi-flow. If $Y > 0$, the PN is said to be conservative[32, 34].

2.2 T-timed discrete PN

T-TDPN is a DPN to which a vector of the minimal duration of transitions firing is added. This latter is denoted as $D_{\min} = (d_{\min\ j})_{j=1,\cdots,q} \in \mathbf{R}^q_{\geq 0}$. Therefore, a T-TDPN system is represented by a triplet $\langle Net, D_{\min}, M_0\rangle$. Each transition t_j^d, $j = 1, \cdots, q$, may fire at a discrete marking M^d and at time τ if it is enabled at M^d and if all tokens required to fire t_j^d have been reserved, at least, from time $\tau - d_{\min\ j}$ (i.e., for a duration not less than $d_{\min\ j}$)[31].

2.3 Untimed continuous Petri nets

Unlike DPN, each continuous transition t_j^c, $j = 1, \cdots, q$ of a continuous PN is enabled at the continuous marking M^c iff $\forall p_i \in {}^\circ t_j^c, m_i^c > 0$. Its enabling degree is no longer restricted to an integer. It is given by (4):

$$\eta(t_j^c, M^c) = \min_{p_i \in {}^\circ t_j^c} \left\{ \frac{m_i^c}{W_{pr}(p_i, t_j^c)} \right\}. \tag{4}$$

Thus, the firing of t_j^c from a real amount α_c such that $0 < \alpha_c \leq \eta(t_j^c, M^c)$ leads to a new marking computed with (2).

Some pioneer works, such as [35], have been performed to study the conservation of important properties through the fluidification. What is crucial to underline is that some properties may be ensured like boundedness and reachability (the reachability space of the DPN is included in the reachability space of the CPN)[35]. However, some others such as liveness and deadlock-freeness can neither be observed nor analyzed once the relaxation is performed because the CPN and its corresponding DPN do not have the same behavior[35]. Consequently, these properties cannot be ensured. As far as liveness is concerned, structural liveness can be ensured but this does not imply liveness[35].

2.4 Timed continuous Petri nets

TCPN is a continuous PN to which a vector denoted by λ, which contains the maximum firing speed of each transition t_j^c, is added such that $\lambda = (\lambda_j)_{j=1,\cdots,q} \in \mathbf{R}^q_{\geq 0}$ and $\lambda[t_j^c] = \lambda_j$. Hence, the TCPN system is given by a triplet $\langle Net, \lambda, M_0 \rangle$. Explicitly, the state equation of such a system is defined by (5):

$$\dot{M}^c(\tau) = W \times F(\tau) \tag{5}$$

where $F(\tau)$ stands for the firing flow such that $F(\tau) = \big(f_j(\tau)\big)_{j=1,\cdots,q}$. Infinite server semantics are considered throughout this paper[31, 33, 36]. Thereby, the maximum flow through each transition t_j^c is written in (6) as the product of the maximum firing speed and the enabling degree. It measures the maximum amount of tokens that can pass through t_j^c per time unit (second (s) or another time unit):

$$f_j(\tau) = \lambda_j \times \eta\big(t_j^c, M^c(\tau)\big) = \lambda_j \times \min_{p_i \in {}^\circ t_j^c} \left\{ \frac{m_i^c(\tau)}{W_{pr}(p_i, t_j^c)} \right\}. \tag{6}$$

The reachability set of TCPN is defined by $R(Net, \lambda, M_0)$. It can be divided into one or several regions as follows: $R(Net, \lambda, M_0) = R_1 \cup \cdots \cup R_z \cup \cdots \cup R_Z$ such that the total number of regions satisfies (7):

$$Z \leq \prod_{j=1}^{q} |{}^\circ t_j^c|. \tag{7}$$

Each region R_z, $z = 1, \cdots, Z$, has a unique configuration represented by a constraint matrix denoted by A_z such that $A_z \in \mathbf{R}^{q\times n}_{\geq 0}$[20]. In order to define explicitly these regions (i.e., the constraint matrices), critical places are defined, they correspond to the subset of places that limit the flow of transitions. In the interior of each region, each transition t_j^c has a single critical place p_i that satisfies (8):

$$\frac{m_i^c(\tau)}{W_{pr}(p_i, t_j^c)} = \eta\big(t_j^c, M^c(\tau)\big). \tag{8}$$

Note that at the borders of regions, several equivalent critical places may exist for a same transition, in that case, only one is selected[21]. Consequently, each row $j = 1, \cdots, q$ of the constraint matrix A_z has only one non-null element in the i-th position:

$$A_z(t_j^c, p_i) \begin{cases} \dfrac{1}{W_{pr}(p_i, t_j^c)}, & \text{if } p_i \text{ is the critical place of } t_j^c \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Thereby, if $M^c(\tau) \in R_z$, the enabling degrees of all transitions can be explicitly expressed as the product $A_z \times M^c(\tau)$. Let Λ be a diagonal matrix such that $\Lambda = \mathrm{diag}(\lambda)$. In a matrix form, the firing flow $F(\tau)$ can be written as (10):

$$F(\tau) = \Lambda \times A_z \times M^c(\tau). \tag{10}$$

Moreover, the state (5) is expressed by (11):

$$\dot{M}^c(\tau) = W \times \Lambda \times A_z \times M^c(\tau). \tag{11}$$

Finally, a linear system is obtained in each region R_z.

2.5 Controlled TCPN

The PN systems, which are subjected to an external control action $U = (u_j)_{j=1,\cdots,n} \in \mathbf{R}^q_{\geq 0}$ applied to the transitions, are called controlled (or forced) PN systems. Throughout this paper, all transitions are assumed to be controlled[37]. In [33], the authors have mentioned that since the transitions model the actuators in real machines or the servers in a station, then the control actions can only slow down or stop the firing flow of the controlled transitions in order to control the system and to reach the steady state configuration. This latter is defined by (M_{ref}, X_{ref}) where M_{ref} corresponds to a desired marking, which is reachable from the initial one M_0, and X_{ref} defines the desired flow. It should be pointed out that the desired configuration should be an equilibrium point[20]. Hence, once it is reached, it should be maintained. Since $\dot{M}(\tau) = W \times X(\tau)$, then M_{ref} and X_{ref} should satisfy the following conditions:

$$\begin{cases} X_{ref} \geq 0 \\ X_{ref} \leq \Lambda \times A_z \times M_{ref} \\ W \times X_{ref} = 0. \end{cases} \tag{12}$$

There exist two control approaches either additive[33] or multiplicative approach[24]. This paper deals with the additive approach for which the controlled flow of TCPN denoted by $X = (x_j)_{j=1,\cdots,q}$ is written, for each transition t_j^c, as: $x_j(\tau) = f_j(\tau) - u_j(\tau)$, where $f_j(\tau)$ is the maximum flow of the transition t_j^c in the corresponding uncontrolled system and $u_j(\tau)$ stands for the applied control action[33]. If $M^c(\tau) \in R_z$, the matrix form of $X(\tau)$ is expressed by (13):

$$X(\tau) = F(\tau) - U(\tau) = \Lambda \times A_z \times M^c(\tau) - U(\tau). \quad (13)$$

Since the flow can only be slowed down and cannot be negative, the control action $U(\tau)$ is dynamically bounded as follows: $0 \leq U(\tau) \leq \Lambda \times A_z \times M^c(\tau)$. Consequently, the overall behavior of the controlled TCPN system, in region R_z, is ruled by the system (14):

$$\begin{cases} \dot{M}^c(\tau) = W \times X(\tau) = W \times \Lambda \times A_z \times M^c(\tau) - W \times U(\tau) \\ 0 \leq U(\tau) \leq \Lambda \times A_z \times M^c(\tau). \end{cases} \quad (14)$$

A sampled time approximation of the continuous time system (14) can be obtained with: $\tau_k = \theta \times k$, where k defines the sampling index, θ denotes the sampling period and τ_k corresponds to the discrete time. Hence, a discrete time approximation of the TCPN is obtained. It will be referred, throughout this article, by $\theta-$TCPN $= \langle Net, \lambda, M_0, \theta\rangle$. Given that $M_0 \geq 0$, the sampling period should be small enough so that any marking $M^c(\tau_k)$ reachable from M_0 is non-negative. It has been proved in [34] that θ must verify:

$$\forall p_i \in P, \sum_{t_j^c \in p_i^\circ} \lambda_j \times \theta < 1. \quad (15)$$

Moreover, let us define $R(Net, \lambda, M_0, \theta)$ as the reachability space of the $\theta-$TCPN. According to [34], as long as θ verifies (15), this set coincides with the reachability space $R(Net, \lambda, M_0)$ of the TCPN. Consequently in order to preserve these properties, it is assumed throughout this paper, the sampling period verifies the inequality given by (15).

At each discrete time τ_k, (16) rules the behavior of the sampled time approximation of (14) in each region R_z:

$$\begin{cases} M^c(\tau_{k+1}) = M^c(\tau_k) + \theta \times W \times X(\tau_k) \\ X(\tau_k) = \Lambda \times A_z \times M^c(\tau_k) - U(\tau_k). \end{cases} \quad (16)$$

Consequently, the marking $M^c(\tau_{k+1})$ can be expressed by (17):

$$M^c(\tau_{k+1}) = B_z \times M^c(\tau_k) - \theta \times W \times U(\tau_k) \quad (17)$$

where the matrix B_z is given by (18):

$$B_z = \theta \times W \times \Lambda \times A_z + I_n \quad (18)$$

and I_n denotes the identity matrices of dimension $n \times n$.

The main purpose of the following sections is to reach and to maintain a desired marking. A new approach, inspired from the MPC, is brought out in order to compute, under some constraints related to the firing flow, the control actions which achieve that goal.

3 Model predictive constant control

The MPC applied to $\theta-$TCPN optimizes a certain cost function that permits the evolution of the system from a current state defined by the initial marking M_0 and its corresponding unforced flow, to the configuration characterized by the desired marking $M_{ref} \in R(Net, \lambda, M_0, \theta)$ and the desired flow X_{ref}. The standard form of the cost function used by the on-line MPC algorithm to control $\theta-$TCPN is given by (19)[34]:

$$\begin{aligned} & J(\tau_{k+N}) = \\ & \quad (M^c(\tau_{k+N}) - M_{ref})^t \times Q_M \times (M^c(\tau_{k+N}) - M_{ref}) + \\ & \quad \sum_{e=0}^{N-1} \Big((M^c(\tau_{k+e}) - M_{ref})^t \times Q_{MI} \times (M^c(\tau_{k+e}) - M_{ref}) + \\ & \quad (X(\tau_{k+e}) - X_{ref})^t \times Q_X \times (X(\tau_{k+e}) - X_{ref}) \Big) \end{aligned} \quad (19)$$

where $(M^c(\tau_{k+N}) - M_{ref})^t \times Q_M \times (M^c(\tau_{k+N}) - M_{ref})$ represents the quadratic error between the continuous marking at the prediction horizon N and the desired one. $\sum_{e=0}^{N-1} (M^c(\tau_{k+e}) - M_{ref})^t \times Q_{MI} \times (M^c(\tau_{k+e}) - M_{ref})$ and $\sum_{e=0}^{N-1} (X(\tau_{k+e}) - X_{ref})^t \times Q_X \times (X(\tau_{k+e}) - X_{ref})$ denote respectively the sum of the quadratic errors between the intermediate markings $M^c(\tau_{k+e})$ and M_{ref}, and between the intermediate flows $X(\tau_{k+e})$ and X_{ref}. Q_M, Q_{MI} and Q_X are positive matrices defining the weight of each term. For simplicity, throughout this paper they are defined by: $Q_M = r_M \times I_n$, $Q_{MI} = r_{MI} \times I_n$ and $Q_X = r_X \times I_q$, where $(r_M, r_{MI}, r_X) \in \mathbf{R}^3_{\geq 0}$.

In [30], an MPC criterion, which takes into account the flow variations, has been introduced in order to reduce the actuator solicitation. The method has been developed for a single step prediction horizon. In the following paragraph, an implicit control scheme which limits the computational complexity and which is suitable for larger prediction horizons, is proposed. It will be demonstrated that such horizons are required to solve control problems with hill-climbing phases (i.e., phases during which the marking moves necessarily away from the reference before it can reach it). Indeed, some configurations are such that to achieve a goal, the current state must first move away from the target. In other words, there does not exist any direct trajectory that systematically decreases the distance between the initial and the target marking. Thus, if the prediction horizon is too small, the obtained markings will be farther than the previous one and consequently the MPC will not converge to the desired configuration. In that case, the horizon of prediction has to be increased to predict "far enough". The problem of hill-climbing phases depends, on the one hand, on the structure of the PN and on the other hand, on the initial and target markings.

Basically, the problem investigated by the authors in [16] and ours are the same. Indeed, the aim is to find a control action U optimizing a certain cost function that drives the system from an initial configuration to a desired one. Both contributions are also motivated by the same difficulty which is the computational complexity that prevents the use of MPC for large systems with large prediction horizons. This complexity results from the piecewise-affine structure of the TCPN that multiplies at each step the number of quadratic problems to be solved. Two major differences between our method and that of [16], must be noticed:

1) The authors in [16] have proposed an explicit MPC scheme that is an alternative to the usual implicit one. It is composed of on-line and off-line parts. As mentioned in [16], the most troublesome procedure is done off-line. Its main idea is to split the controllable set into polytopes described by linear inequalities in which the control action U is described as a piecewise affine function of the state. All linear and quadratic programs are solved off-line and then the on-line part applies the adequate feedback depending on the polytope that the marking belongs to. In contrast, the approach, proposed throughout this paper, is mainly based on the implicit scheme, it includes some simplifications. First, the intermediate markings and flows are no longer taken into account in the expression of the cost function J. Second, the control action is assumed to be constant over the prediction horizon (but it changes from one step to another). Third, the prediction horizon N is updated on-line in a given range. These simplifications reduce the exponential explosion of region switches.

2) The hill-climbing phase problem has not been investigated in [16]. In such problem, a large prediction horizon is recommended since a small one does not predict "far enough" to detect the fact that the cost function J will be reduced after climbing. Consequently, it is necessary to increase N until the value that ensures the convergence of the system to the desired configuration. Therefore, if the explicit or the usual implicit MPC are considered to resolve the hill-climbing phase problem, we will face two major difficulties. As far as the explicit MPC is considered, the computational complexity of the off-line part will highly increase and it can be prohibitive for certain values of N[16]. In addition, if the modeled system has many regions, the on-line part will encounter a serious difficulty to choose the adequate feedback to be applied. Concerning the usual implicit MPC, the control scheme can no longer be implementable for large prediction horizons since the computation time required to solve the quadratic problem for one iteration is larger than the sampling period. Thereby, the implicit and the explicit schemes are not applicable to complex systems[16].

In [27], the control method is also based on the MPC. Fundamentally, the control design problems investigated by the authors in [27] and ours are quite different. Obviously, both papers use the MPC controller and try to improve asymptotic stability. But we use dissimilar concepts to reach the aims. First of all, in [27], the authors have assumed that the system is functionally distributed and composed of several subsystems interconnected by means of some places that model buffer in the original system. In contrast, in the proposed paper, non-distributed systems have been considered. Secondly, in [27], the aim is, on one side, to drive all the subsystems from a current marking to their desired one (the desired flow has not been taken into account) and on the other side, to ensure the asymptotic stability of the system. In this regard, the authors have proposed a centralized MPC controller and they have assumed that, during the prediction, the obtained markings belong to a closed convex subset of the reachability space. Then, they have focused on the local control of distributed systems and they have applied the developed MPC controller as a local controller for each subsystem. However, in the proposed control method, the desired configuration is a steady state composed of desired marking and flow. Besides, the MPCC, with constant control actions during the prediction, reduces the computational complexity. The adaptive prediction horizon solves the problem of the hill-climbing-phases. To ensure stability, we suppose that, in the β_neighborhood of the desired marking, a terminal constraint is added. This constraint ensures that the marking belongs to a straight line $M^c(\tau_k) - M_{ref}$[34].

3.1 Model predictive constant control for θ-TCPN

In order to avoid the computational complexity, the combinatorial explosion problem of the constraints and of the search space, the idea is to develop MPC with constant control sequences over an adaptive prediction horizon that depends on the variation of J and on the region switches. The proposed approach is called MPCC. The considered problem is to minimize the cost function (20) which is a simplified form of (19):

$$J(\tau_{k+N}) = J_M(\tau_{k+N}) + J_X(\tau_{k+N-1}) \tag{20}$$

where $J_M(\tau_{k+N})$ and $J_X(\tau_{k+N-1})$ are respectively defined by (21) and (22):

$$J_M(\tau_{k+N}) = \left(M^c(\tau_{k+N}) - M_{ref}\right)^t \times Q_M \times \left(M^c(\tau_{k+N}) - M_{ref}\right) \tag{21}$$

$$J_X(\tau_{k+N-1}) = \left(X(\tau_{k+N-1}) - X_{ref}\right)^t \times Q_X \times \left(X(\tau_{k+N-1}) - X_{ref}\right) \tag{22}$$

so that at each prediction step $e = 1, \cdots, N$, the flow $X(\tau_{k+e-1})$ must verify the following constraints:

$$0 \leq X(\tau_{k+e-1}) \leq F(\tau_{k+e-1}) \tag{23}$$

where $F(\tau_{k+e-1})$ is the corresponding unforced flow.

The control actions are assumed to be constant over the prediction horizon. Consequently, taking into account the intermediate markings and flow will not change significantly the computation of U and the obtained results in both cases will be almost the same. Moreover, it should be pointed that even if the proposed approach computes a control action U such that the intermediate markings are not taken into consideration in the cost function J, they satisfy all constraints (in term of region and of positivity of the firing speeds).

Proposition 1 shows that the optimization of cost function (20) under the constraints (23) may be transformed into a quadratic optimization problem[38].

Proposition 1. Let $\langle Net, \lambda, M_0, \theta\rangle$ be a $\theta-$TCPN system such that all the transitions are controllable. Let $N > 0$ and assume that (M_{ref}, X_{ref}) is a reachable configuration that verifies (12) and that the marking $M^c(\tau_k)$ stays in the same region R_z during the interval $[\tau_k,\ \tau_{k+N}[$. Then, the constant control $U(\tau_k)$ that minimizes the cost function (20), under the constraints (23), is the solution of the quadratic optimization problem (24) under the constraints (25):

$$U(\tau_k)=\arg\min_U\left\{\frac{1}{2}\times U(\tau_k)^t\times\Omega\times U(\tau_k)+\rho(\tau_k)^t\times U(\tau_k)\right\} \tag{24}$$

$$\forall e=1,\cdots,N,\quad \begin{array}{l}\text{a)}S_{e-1}\times U(\tau_k)\geq 0\\ \text{b)}S_{e-1}\times U(\tau_k)\leq\Lambda\times A_z\times(B_z)^{e-1}\times M^c(\tau_k).\end{array} \tag{25}$$

Note that the inequalities (25) are considered element-wise. Ω and $\rho(\tau_k)$ are respectively defined by (26) and (27):

$$\Omega=\theta^2\times W^t\times\Sigma_N^t\times Q_M\times\Sigma_N\times W+S_{N-1}^t\times Q_X\times S_{N-1} \tag{26}$$

$$\begin{aligned}\rho(\tau_k)=\theta\times W^t\times\Sigma_N^t\times Q_M\times(M_{ref}-B_z^N\times M^c(\tau_k))+\\ S_{N-1}^t\times Q_X\times(X_{ref}-\Lambda\times A_z\times(B_z)^{N-1}\times M^c(\tau_k))\end{aligned} \tag{27}$$

such that B_z is defined by (18) and:

$$\Sigma_0=0,\quad \Sigma_e=\sum_{j=0}^{e-1}(B_z)^j,\quad \forall e=1,\cdots,N \tag{28}$$

$$S_e=\theta\times\Lambda\times A_z\times\Sigma_e\times W+I_q,\forall e=0,\cdots,N. \tag{29}$$

Proof. Under the assumption that, the control action is constant and the markings stay in the same region R_z during the interval $[\tau_k,\ \tau_{k+N}[$, the markings $M^c(\tau_{k+e})$ and the forced flows $X(\tau_{k+e-1})$, $\forall e=1,\cdots,N$ are iteratively obtained from (16). They are respectively expressed by (30) and (31):

$$M^c(\tau_{k+e})=B_z^e\times M^c(\tau_k)-\theta\times\Sigma_e\times W\times U(\tau_k) \tag{30}$$

$$X(\tau_{k+e-1})=\Lambda\times A_z\times B_z^{e-1}\times M^c(\tau_k)-S_{e-1}\times U(\tau_k). \tag{31}$$

When $e=N$, $M^c(\tau_{k+e})$ and $X(\tau_{k+e-1})$ are used to rewrite the criteria $J_M(\tau_{k+N})$ and $J_X(\tau_{k+N})$. All constant terms with regard to $U(\tau_k)$ are removed. Thereby, simplified forms of these criteria are introduced as $J_{M,U}(\tau_{k+N})$ and $J_{X,U}(\tau_{k+N-1})$ which are given by (32) and (33):

$$\begin{aligned}J_{M,U}(\tau_{k+N})=&\theta^2\times U(\tau_k)^t\times W^t\times\Sigma_N^t\times Q_M\times\Sigma_N\times W\times\\ &U(\tau_k)+2\times\theta\times\left(M_{ref}-B_z^N\times M^c(\tau_k)\right)^t\times\\ &Q_M\times\Sigma_N\times W\times U(\tau_k)\end{aligned} \tag{32}$$

$$\begin{aligned}J_{X,U}(\tau_{k+N-1})=&U(\tau_k)^t\times S_{N-1}^t\times Q_X\times S_{N-1}\times U(\tau_k)+\\ &2\times\left(X_{ref}-\Lambda\times A_z\times B_z^{N-1}\times M^c(\tau_k)\right)^t\times\\ &Q_X\times S_{N-1}\times U(\tau_k).\end{aligned} \tag{33}$$

Hence, the cost function is reduced to $J_U(\tau_{k+N})$ (34) which depends on the control action $U(\tau_k)$:

$$J_U(\tau_{k+N})=\frac{1}{2}\times\left(J_{M,U}(\tau_{k+N})+J_{X,U}(\tau_{k+N-1})\right). \tag{34}$$

$J_U(\tau_{k+N})$ is rewritten by (35):

$$J_U(\tau_{k+N})=\frac{1}{2}\times U(\tau_k)^t\times\Omega\times U(\tau_k)+\rho(\tau_k)^t\times U(\tau_k) \tag{35}$$

such that the terms Ω and $\rho(\tau_k)$ are defined by (26) and (27). Given that, $\forall e=1,\cdots,N$, $F(\tau_{k+e-1})=\Lambda\times A_z\times B_z^{e-1}\times M^c(\tau_k)$, the constraints (25) on the control action $U(\tau_k)$ are obtained by substituting, $X(\tau_{k+e-1})$ by (31) in (23). □

Note that the optimization problem (24) is always feasible, for any $M^c(\tau_k)\geq 0$, as it admits at least one solution ($U(\tau_k)=\begin{bmatrix}0&\cdots&0\end{bmatrix}^t$ is a trivial solution that satisfies all the constraints given by (25)). Notice that $\forall e=1,\cdots,N, F(\tau_{k+e-1})=\Lambda\times A_z\times B_z^{e-1}\times M^c(\tau_k)\geq 0$. Asymptotic convergence and stability of MPC can be obtained by adding a terminal constraint (TC)[39]. This method has been adapted for $\theta-$TCPN in order to ensure the convergence of the system toward the desired marking[34]. The terminal constraint ensures that the continuous marking $M^c(\tau_{k+1})$ belongs to the straight line from $M^c(\tau_k)$ to M_{ref}[34]. It is given by (36):

$$\begin{aligned}M^c(\tau_{k+1})=M^c(\tau_k)+\alpha\times(M_{ref}-M^c(\tau_k))\\ \text{e.g.,}\quad 0<\alpha\leq 1.\end{aligned} \tag{36}$$

Let us first define a *δ_neighborhood* of the desired marking as an area for which $J_M(\tau_k)\leq\delta$, where $\delta>0$. In that neighborhood, if J decreases, N will also decrease. The marking reaches a β_neighborhood when $N=1$ ($\beta\leq\delta$). In this last phase of the trajectory, the linear constraint, given by (36), is introduced as long as the desired configuration is not obtained (i.e., $J(\tau_k)>\epsilon$). In contrast, the terminal constraint is not used during the first part of the trajectory when N is larger than 1. Consequently, the constraint to be satisfied by the control actions is expressed as (37):

$$\begin{aligned}\theta\times W\times U+\alpha\times\left(M_{ref}-M^c(\tau_k)\right)=(B_z-I_n)\times M^c(\tau_k)\\ \text{e.g.,}\quad 0<\alpha\leq 1.\end{aligned} \tag{37}$$

The Lyapunov stability implies that solutions that start close enough to the equilibrium point remain also close enough. Asymptotic stability includes Lyapunov stability but it requires that the solutions converge to the equilibrium. Hence, to prove the asymptotic stability of the TCPN, a Lyapunov function must be found, and then it must be proved that it is a strictly decreasing function.

Proposition 2 proves the asymptotic stability of the closed-loop system thanks to the use of the terminal constraint.

Proposition 2. Let $\langle Net,\lambda,M_0,\theta\rangle$ be a θ-TCPN system such that all transitions are controllable. Let (M_{ref},X_{ref}) be a reachable configuration that verifies (12). Assuming that there exists an instant τ_k from which the β_neighborhood is reached ($N=1$), then the closed-loop system is asymptotically stable.

Proof. If $N=1$, the terminal constraint (36) is applied and the marking $M^c(\tau_{k+1})$ is forced to belong to the straight line $M^c(\tau_k) - M_{ref}$. Equation (36) is rewritten as: $M^c(\tau_{k+1}) - M_{ref} = M^c(\tau_k) + \alpha \times \big(M_{ref} - M^c(\tau_k)\big) - M_{ref}$ which leads to: $M^c(\tau_{k+1}) - M_{ref} = (1-\alpha) \times \big(M^c(\tau_k) - M_{ref}\big)$. Hence, $J_M(\tau_{k+1})$, defined by (21), is reformulated as:

$$J_M(\tau_{k+1}) = (1-\alpha)^2 \times J_M(\tau_k). \tag{38}$$

If $0 < \alpha \leq 1$, the criterion $J_M(\tau_k)$ could be used as a Lyapunov function, since it is defined positive and strictly decreasing (i.e., $J_M(\tau_{k+1}) < J_M(\tau_k)$). □

The previous results are used to propose a control design encoded in Algorithm 2. Let N_0 and $N_{\max}$ be respectively the initial and the maximum values of the prediction horizon N. The main Algorithm 1, which controls the $\theta-$TCPN to reach the desired configuration, consists of computing, at each sampling step, the control action $U(\tau_k)$ that minimizes (35) under the constraints (25) such that the marking stays in the same region. In order to alleviate possible hill-climbing phases, N may grow, during the iterations, until $N_{\max}$, as long as the cost function does not decrease (i.e., $J(\tau_k) - J(\tau_{k+N}) \leq 0$) and while the first predicted $N-1$ markings stay in the same region. When the current marking $M^c(\tau_k)$ reaches a δ_neighborhood of M_{ref} and as long as the cost function J is strictly decreasing, the prediction horizon should decrease. This procedure has been added in order to decrease the prediction horizon to $N = 1$ and to promote the use of the terminal constraint in the β_neighborhood.

Algorithm 1. Control of $\theta-$TCPN using MPCC (Inputs: M_0, N_0, δ, ϵ. Outputs: U and M^c)

```
Parameter initialization: $\tau_k \leftarrow 0$, $M^c(\tau_k) \leftarrow M_0$,
  $N \leftarrow N_0$, $TC \leftarrow 0$, $M^c \leftarrow M^c(\tau_k)$, $U \leftarrow \varnothing$
Compute $J(\tau_k)$
While $J(\tau_k) > \epsilon$
  Compute $U(\tau_k)$ using Algorithm 2
  $U \leftarrow U \cup U(\tau_k)$
  Compute $X(\tau_k)$, $M^c(\tau_{k+1})$ and $J(\tau_{k+1})$
  $M^c \leftarrow M^c \cup M^c(\tau_{k+1})$
  If $J_M(\tau_{k+1}) \leq \delta$, then
    If $J(\tau_{k+1}) < J(\tau_k)$, then
      $N \leftarrow \max(N-1, 1)$
    End if
    If $N = 1$, then
      $TC \leftarrow 1$
    End if
  End if
  $\tau_k \leftarrow \tau_{k+1}$
End while
```

Algorithm 2. MPCC quadratic optimization (Inputs: $M^c(\tau_k)$, N, $N_{\max}$ and $J(\tau_k)$. Outputs: $U(\tau_k)$ and N)

```
Parameter initialization: $N_{limit} \leftarrow N_{\max}$, $flag \leftarrow 0$
While $flag = 0$
  $flag \leftarrow 1$
  Determine the region $R\big(M^c(\tau_k)\big)$ of $M^c(\tau_k)$
  If $TC = 0$, then
    Compute $U(\tau_k)$ that minimizes (24) under
     the constraints (25)
    Compute $J(\tau_{k+N})$
    Determine the region $R\big(M^c(\tau_{k+e})\big)$ of the markings
     $M^c(\tau_{k+e})$, $\forall e = 1, \cdots, N-1$
    If $\exists e < N-1$ such that
     $R\big(M^c(\tau_{k+e})\big) \neq R\big(M^c(\tau_k)\big)$, then
      $N_{limit} \leftarrow e$
      $N \leftarrow e$
      $flag \leftarrow 0$
    Else if $J(\tau_k) - J(\tau_{k+N}) \leq 0$ and $N < N_{limit}$
    then
      Increase $N$
      $flag \leftarrow 0$
    End if
  Else if
    Compute $U(\tau_k)$ that minimizes (24)
     under the constraints (25) and (36)
  End if
End while
```

3.2 Computational complexity

In order to compare our approach with the existing ones and in particular with [16], Tables 1−4 determine, for each case, the resulting computational efforts for a prediction horizon with N steps. Different cases are distinguished according to the definition of the performance criterion J (the standard form 20) in Tables 3 and 4 and the simplified one (19) in Tables 1 and 2, to the region switches (that could be considered or not during the prediction) and finally to the control action U (that may be constant or not). The complexity is evaluated wrt the number of variables, the number of constraints and wrt the number of required elementary operations to compute the cost function over the considered prediction horizon.

One may observe that when the control action is variable, the number of variables is multiplied by N and taking into account a changing of regions increases the complexity by a factor Z^{N-1} (where Z represents the number of regions). We draw the attention of the readers that our approach presents a lower cost than that of [16].

The parameter $N_{\max}$ is extremely important in the predictive control. It should be carefully chosen in order to ensure that the computational time required at each step by the MPCC does not exceed the sampling period of the controller. For that reason, Proposition 3 provides an upper-bound for this parameter. Consequently this result ensures that the approach is consistent with real-time constraints.

Proposition 3. Let $\langle Net, \lambda, M_0, \theta\rangle$ be a $\theta-$TCPN system such that all transitions are controllable. The maximal value $N_{\max}$ of the prediction horizon N should satisfy:

$$N_{\max} \leq \left\lfloor \frac{\theta}{\Psi(1)} \right\rfloor \tag{39}$$

where $\Psi(1)$ is the computational time required to solve the quadratic problem (24) under the constraints (25), for a prediction horizon of size $N = 1$.

Proof. In our approach, the intermediate markings and flows are not considered for the computation of the cost function and the control action is supposed to be constant over the prediction horizon. Moreover, the marking is not supposed to leave the current region. In that case, the complexity wrt the constraints is $\mathrm{O}(N \times q)$ where q is the number of control actions and wrt the cost function is $\mathrm{O}(N \times (n+q))$ where n is the dimension of the marking. Consequently, the complexity of the cost function is linear wrt the prediction horizon. Thereby, the time required to compute the cost function, at each time step τ_k of Algorithm 1 is linear wrt the prediction horizon. For $N_{\max}$ prediction steps, it is given by $\Psi(N_{\max}) = \Psi(1) \times N_{\max}$. The MPCC is implementable for $N_{\max}$ if the computational time for one iteration (i.e., $\Psi(N_{\max})$) is smaller than the sampling period θ. Thus, the inequality (39) is obtained. □

3.3 Example

Let us consider the TCPN system shown in Fig. 1 such that the initial marking and the firing rate vector are respectively $M_0 = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^t$ and $\lambda = \begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix}^t$:

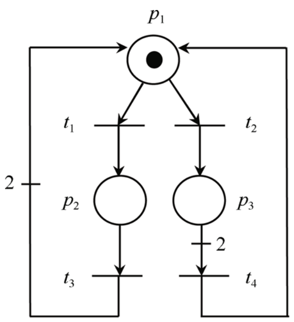

Fig. 1 Example of PN system with $M_0 = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^t$ (PN1)

Two cycles can be distinguished, namely if the transition t_1 is fired and t_2 is blocked a token producer cycle is triggered. Otherwise, a token consumer cycle starts. Numerical simulations have been performed using the following values: the desired configuration is $\left(M_{ref} = \begin{bmatrix}3 & 0 & 0\end{bmatrix}^t, X_{ref} = \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}^t\right)$ the weighting factors are $r_M = 1$ and $r_X = 0.1$ (in order to give more importance to the marking with regard to the flow) and the sampling period is $\theta = 0.2\,\mathrm{s}$ (the maximum value that verifies the inequality given by (15)). Finally, the initial and the maximum prediction horizons satisfy respectively $N_0 = 1$ and $N_{\max} = 25$. The marking trajectory, the performance criterion, the marking and the control action variations are respectively given by Figs. 2−5.

Fig. 2 shows that M_{ref} cannot be reached directly from M_0. At first, the marking trajectory moves away from the desired marking, then from $\tau_k \approx 2.6\,\mathrm{s}$, it comes closer to the target. Therefore, to decrease the performance criterion J, a transient increase of J (hill-climbing phase shown in Fig. 5) is required. The only solution to compute correctly the control action is to predict the marking over an horizon larger than the hill-climbing phase. The adaptive prediction horizon given by Algorithm 2, at each instant τ_k is $N = 13$ which exactly corresponds to the duration in which the marking approaches M_{ref}. From $\tau_k \approx 9.4\,\mathrm{s}$,

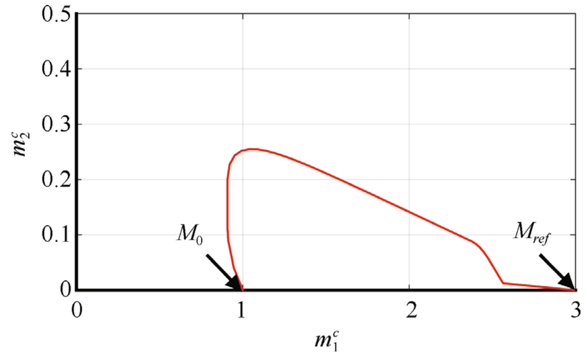

Fig. 2 Marking trajectory of PN1

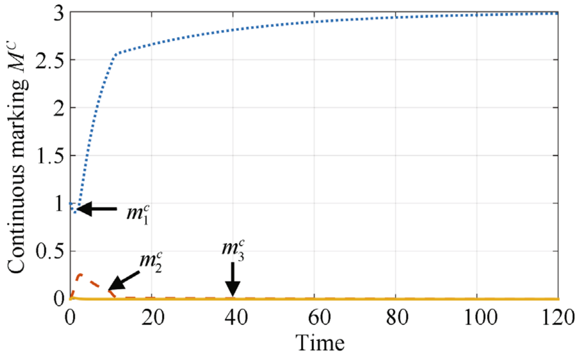

Fig. 3 Marking variations of PN1

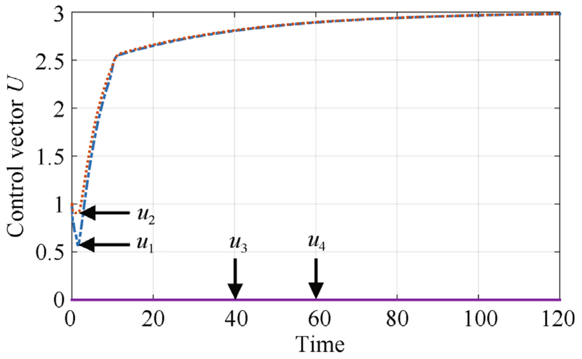

Fig. 4 Control action of PN1

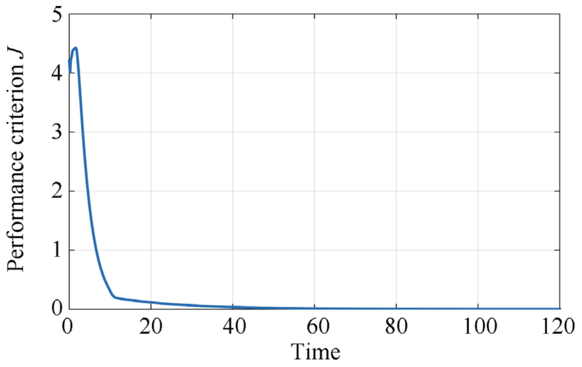

Fig. 5 Performance criterion of PN1

Table 1 Complexity of cost function (19) without changing of region

	U constant (our approach)	U variable
Number of variables	q	$N \times q$
Complexity wrt constraints	$\mathrm{O}(N \times q)$	$\mathrm{O}(q \times N)$
Complexity wrt the cost function	$\mathrm{O}(N \times (n+q))$	$\mathrm{O}(N \times (n+q))$

Table 2 Complexity of cost function (19) with changing of region

	U constant	U variable
Number of variables	q	$N \times q$
Complexity wrt constraints	$\mathrm{O}(Z^{N-1} \times N \times q)$	$\mathrm{O}(Z^{N-1} \times q \times N)$
Complexity wrt the cost function	$\mathrm{O}(Z^{N-1} \times N \times (n+q))$	$\mathrm{O}(Z^{N-1} \times N \times (n+q))$

Table 3 Complexity of cost function (20) without changing of region

	U constant	U variable
Number of variables	q	$N \times q$
Complexity wrt constraints	$\mathrm{O}(N \times q)$	$\mathrm{O}(q \times N)$
Complexity wrt the cost function	$\mathrm{O}(N^2 \times (n+q))$	$\mathrm{O}(N^2 \times (n+q))$

Table 4 Complexity of cost function (20) with changing of region

	U constant	U variable (approach [16])
Number of variables	q	$N \times q$
Complexity wrt constraints	$\mathrm{O}(Z^{N-1} \times N \times q)$	$\mathrm{O}(Z^{N-1} \times q \times N)$
Complexity wrt the cost function	$\mathrm{O}(Z^{N-1} \times N^2 \times (n+q))$	$\mathrm{O}(Z^{N-1} \times N^2 \times (n+q))$

the δ_neighborhood of the desired marking has been reached. Consequently, the prediction horizon starts to decrease until $\tau_k \approx 11.6$ s. In that instant, the β_neighborhood has been reached ($N = 1$) and thereby the terminal constraint has been applied which manifests by a linear continuous trajectory shown in Fig. 2.

Let us point out that the MPC and the MPCC are two constrained control methods. Hence, some properties like boundedness and deadlock-freeness can be ensured by adding suitable constraints. As far as the deadlock-freeness is concerned, assuming that the list of deadlocks is already known, then it can be avoided during the optimization. Each deadlock can be encoded with a set of linear constraints on the marking. Adding these constraints in the MPCC makes the deadlocks forbidden. The constrained MPCC will act in a similar way as the monitor places used in supervisory control methods[13]. Note that fluidification does not preserve the deadlock-freeness in the general case. For this reason, we prefer to keep the assumption of deadlock-free T-TDPN in the following section. Furthermore, boundedness may be also guaranteed by adding linear constraints to upper bound the marking of each place.

4 Model predictive constant control for T-TDPN

The motivation of this part is to determine a control sequence of sub-minimal duration, which allows a T-TDPN to reach a desired marking, without exploring the reachability graph that may suffer from state explosion problem.

When the nets are heavily loaded, $\theta-$TCPN systems have been proved to be adequate approximation of T-TDPN[18]. For weakly loaded nets, the approximation may be correct (it is not always the case[40]). In this section, it is assumed that the $\theta-$TCPN approximation is acceptable. Then, the control sequence of the T-TDPN will be deduced from the corresponding $\theta-$TCPN controlled via the MPCC.

The proposed approach is suitable to solve scheduling problems, in particular when makespan optimization is concerned[11].

4.1 MPCC for T-TDPN

In this section, only deadlock-free T-TDPN with non-immediate transition firings ($d_{\min\, j} \neq 0, \forall j = 1, \cdots, q$) are considered. All discrete transitions are assumed to be controlled. Consequently, at each instant τ_k, the enabled discrete transitions, at the current discrete marking M^d, can be enforced or avoided. In other words, they are not necessarily fired (i.e., the firing of some discrete transitions can be delayed).

The main idea is to design the control sequence of the T-TDPN according to the control actions generated from its equivalent $\theta-$TCPN (same net structure, same initial marking, same time specifications and same control goal). Indeed, a T-TDPN system $\langle Net, D_{\min}, M_0\rangle$ and its continuous approximation defined by $\langle Net, \lambda, M_0, \theta\rangle$, are considered such that $\lambda_j = \frac{1}{d_{\min\ j}}$. The desired marking $M_{ref} \in \mathbf{N}^n_{\geq 0}$ is assumed to be reachable from the initial one M_0.

For each instant τ_k, the MPCC will be performed on the $\theta-$TCPN in order to compute the control action $U(\tau_k)$, which minimizes the cost function $J_U(\tau_{k+N})$ under the constraints given by (25) and (36). Then, the instantaneous flow $X(\tau_k)$ is computed. The cumulative flow multiplied by θ, denoted as $X_{\rm sum}(\tau_k) = (x_{{\rm sum}j}(\tau_k))_{j=1,\cdots,q}$ and given by (40), corresponds to the number of all tokens that have moved through each transition t^c_j until the instant τ_k:

$$X_{\rm sum}(\tau_k) = \theta \times \sum_{i=0}^{k} X(\tau_i). \tag{40}$$

The link between the T-TDPN and the $\theta-$TCPN is made from the cumulative flow point of view. Hence, when a component of $X_{\rm sum}$ reaches 1 for a given continuous transition t^c_j (i.e., $x_{{\rm sum}\ j}(\tau_k) = 1$), this means that one token, at least, has moved through the equivalent discrete transition t^d_j in the T-TDPN. Two cases may occur; first, t^d_j is not enabled at the current marking M^d; in that case no solution could be found, the method fails. Second, t^d_j is enabled and is fired.

Let Φ be the quadratic distance between the continuous and the discrete markings. It is defined by (41):

$$\Phi = (M^d - M^c)^t \times (M^d - M^c). \tag{41}$$

This criterion will be evaluated for each enabled discrete transition. Let t^{d_opt} be the discrete transition that minimizes ϕ (i.e., the resulting discrete marking M^d moves closer to the continuous one M^c). Two other sub-cases have to be distinguished. On the one hand, if the transitions t^d_j corresponds to transition t^{d_opt}, then the component of $X_{\rm sum}$ that corresponds to t^d_j (t^c_j in the $\theta-$TCPN) is reset to zero. On the other hand (if t^d_j does not correspond to t^{d_opt}), the continuous trajectory is cut at M^c, a new continuous trajectory is initialized at the marking M^d that results from the firing of t^d_j (the initial continuous marking is assumed to be equal to M^d) and the vector $X_{\rm sum}$ is reset to zero.

Note that, sometimes, there is more than one component of $X_{\rm sum}$ that reaches 1. In that case, there is also more than one possibility for the transition t^d_j. Thereby, the enabled one that minimizes the criterion Φ is selected and assigned to t^d_j.

Consequently, the proposed approach, encoded in Algorithm 3, ensures that the discrete marking is as close as possible to the continuous trajectory, as detailed in Proposition 4. Let us point out that Algorithm 3 gives an ordered control sequence σ of length H and a sequence M^d_{tot} of the successive discrete markings.

Algorithm 3. Control of T-TDPN using MPCC (Inputs: M_0, N_0, δ and ϵ. Outputs: σ, H, M^d_{tot})

```
Parameter initialization: $\tau_k \leftarrow 0$, $h \leftarrow 0$, $M^c(\tau_k) \leftarrow$
    $M_0$, $\sigma \leftarrow \varnothing$, $N \leftarrow N_0$, $flag \leftarrow 1$, $M^d_{tot}(h) \leftarrow$
    $M^d$, $M^d_{tot} \leftarrow \varnothing$, $TC \leftarrow 1$, $X_{\rm sum} \leftarrow 0$
Compute $X(\tau_k)$ and $J(\tau_k)$
While $J(\tau_k) > \epsilon$ and $flag = 1$
  Compute the control action $U(\tau_k)$ using Algorithm 2
  Compute $X(\tau_k)$, $X_{\rm sum}(\tau_k)$, $M^c(\tau_{k+1})$ and $J(\tau_{k+1})$
  If $\exists t^c_j$ such that $x_{{\rm sum}\ j}(\tau_k) \geq 1$, then
    $h \leftarrow h + 1$
    Determine the discrete transition $t^d_j$
    If $t^d_j$ is enabled at $M^d_{tot}(h-1)$, then
      $flag \leftarrow 1$
      Compute the new discrete marking $M^d$
      $M^d_{tot} \leftarrow M^d_{tot} \cup M^d$
      $\sigma \leftarrow \sigma \cup t^d_j$
      Determine the optimal discrete transition $t^{d\_opt}$
      If $t^d_j = t^{d\_opt}$, then
        $x_{{\rm sum}\ j} \leftarrow 0$
      Else
        Reinitialize the continuous trajectory by $M^d$
        (i.e., $M^c(\tau_{k+1}) \leftarrow M^d_{tot}(h)$)
        Reset $X_{\rm sum}(\tau_k)$ to zero
      End if
    Else
      $flag \leftarrow 0$
    End If
  End If
  If $J_M(\tau_{k+1}) \leq \delta$, then
    If $J(\tau_{k+1}) < J(\tau_k)$, then
      $N \leftarrow \max(N-1, 1)$
    End if
    If $N = 1$, then
      $TC \leftarrow 1$
    End if
  End if
  $\tau_k \leftarrow \tau_{k+1}$
End while
If $flag = 1$, then
  $H \leftarrow h$
End if
```

Notice that the firing of each transition $t^d_j \in \sigma$ respects the temporal constraints given by $D_{\min}$. In other words, the firing occurs as soon as possible. Then, the firing dates are computed and stored in the temporal control sequence σ^{timed}.

The obtained control sequence σ leads, most of the time, to M_{ref}. Indeed, at any step $h = 1, \cdots, H$, the continuous marking can be written by $M^c = M^d_0 + W \times X_{\rm sum}$ where M^d_0 stands for the last discrete marking which was used to reinitialize the continuous trajectory and $X_{\rm sum}$ is considered since that last reset. This latter can be expressed as: $X_{\rm sum} = \lfloor X_{\rm sum} \rfloor + \{X_{\rm sum}\}$ such that $\lfloor\cdot\rfloor$ and $\{\cdot\}$ denote respectively the integer and the fractional parts of $(\cdot)$. Note that $\lfloor X_{\rm sum} \rfloor$ corresponds to the firing vector of the T-TDPN. Consequently, $M^d = M^d_0 + W \times \lfloor X_{\rm sum} \rfloor = M^c - W \times \{X_{\rm sum}\}$. When $\theta-$TCPN reaches the desired configuration, $M_c = M_{ref}$ and $\{X_{\rm sum}\} = X_{ref}$. Then, the fundamental equation of the T-TDPN is given by $M^d = M_{ref} - W \times X_{ref}$ with $W \times X_{ref} = 0$. Consequently, $M^d = M_{ref}$.

Proposition 4 gives some properties satisfied by the discrete trajectory of the T-TDPN which is obtained according to the MPCC approach. To the best of our knowledge, it is the first attempt to derive directly a discrete MPC schema from a continuous one.

Proposition 4. Let $\langle Net, D_{\min}, M_0\rangle$ be a deadlock-free T-TDPN system such that all transitions are controlled and non-immediate. Let $\langle Net, \lambda, M_0, \theta\rangle$ be its discrete time continuous approximation such that $\lambda_j = \frac{1}{d_{\min j}}$. $M_{ref} \in \mathbf{N}^n_{\geq 0}$. (M_{ref}, X_{ref}) is supposed to be a reachable configuration that verifies (12). If Algorithm 3 returns a sequence M^d_{tot} of discrete markings that converges to M_{ref}, then the following properties hold:

1) $\forall h = 1, \cdots, H$, the discrete marking $M^d_{tot}(h-1)$ is reachable from $M^d_{tot}(h)$ and the discrete trajectory $M_0\,[\sigma(1) > M^d_{tot}(1), \cdots, M^d_{tot}(H-1)\,[\sigma(H) >$ is feasible from M_0 where $\sigma(h)$ stands for the h-th discrete transition of the sequence σ.

2) $\forall h = 1, \cdots, H$, $M^d_{tot}(h)$ is the closest discrete marking from M^c.

Proof.

1) $\forall h = 1, \cdots, H$, $\sigma(h)$ is selected among the transitions which are enabled at $M^d_{tot}(h-1)$. By construction the trajectory is feasible.

2) According to Algorithm 3, $\sigma(h)$ corresponds to the transition t^c_j such that $x_{\text{sum } j} = 1$. If the resulting discrete marking $M^d_{tot}(h)$ minimizes Φ, then property 2 is satisfied. Otherwise, the continuous trajectory is reinitialized by $M^d_{tot}(h)$ and the property 2 is also satisfied. □

At this stage, it appears interesting to point out some differences and limitations of the proposed approach in comparison with approaches, in particular, those proposed in [1 − 6]. In [5, 6], a heuristic search based on some extensions of the A* algorithm with various heuristic functions is studied in order to design control sequences for flexible manufacturing systems represented with a particular subclass of T-TDPN named Buffer-nets. These latter nets exclude the modeling of all assembling and disassembling operations because the number of input and output buffers is limited to one for each transition. This limitation does not hold with the proposed approach. However, the main limitation in [5, 6] concerns the search method by itself. Indeed, the performance of the heuristic approach strongly depends on the choice of the function that estimates the makespan. In [5], good results are provided because it is assumed that the selected function is the optimal one, that supposes to solve first the problem and so the method is no longer required. In [6], improved heuristic functions based on a resource cost reachability matrix are proposed. Such functions provide good results with an acceptable complexity as long as the complete system is considered as a DES. Hence, this approach concerns only discrete event systems and cannot be applied with fluid models. Concerning [1, 2, 4], the authors are interested in the synthesis of a PN-based supervisor to model automated manufacturing systems (AMS) using only particular structures of the DPN such as feedback system of sequential systems with shared resources (FS^4R) or system of simple sequential processes with resources (S^3PR) whereas all deadlock-free DPN are concerned by our proposed method. Besides, their main purposes concerned liveness and deadlock-avoidance. The synthesis of such a supervisor is based on a partial reachability analysis due to a distributed control technique. Then, they have extended their results to large scale AMS where the problem is simplified by restricting the behavior of the concerned DPN to live trajectories and deadlock-free. In other words, the proposed algorithm has to compute only acceptable trajectories that lead to the desired destination[2]. In our approach, deadlock-freeness assumption is made concerning the T-TDPN and the developed control has to minimize a given cost function and is realized thanks to the corresponding TCPN, obtained through fluidification. In both papers the maximal permissiveness is not ensured.

4.2 Example

In order to illustrate the proposed method, let us consider the T-TDPN shown in Fig. 6 with $M_0 = \begin{bmatrix}2 & 4 & 1 & 4\end{bmatrix}^t$, $D_{\min} = \begin{bmatrix}1 & 1 & 1\end{bmatrix}^t$ and $\lambda = \begin{bmatrix}1 & 1 & 1\end{bmatrix}^t$.

Numerical simulations have been performed using the following values: the desired configuration is $\left(M_{ref} = \begin{bmatrix}7 & 0 & 0 & 3\end{bmatrix}^t, X_{ref} = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^t\right)$, $r_M = 1$, $r_X = 0.1$ (as in the previous example), $\theta = 0.1$ s, $N_0 = 1$ and $N_{\max} = 10$. The quadratic marking error J_M, X_{sum} and the marking trajectories (continuous and discrete one) are respectively given by Figs. 7−9.

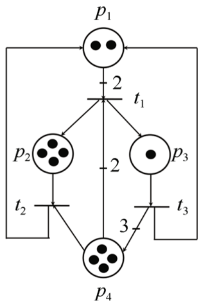

Fig. 6 Example of deadlock-free PN system with $M_0 = \begin{bmatrix}2 & 4 & 1 & 4\end{bmatrix}^t$ (PN2)

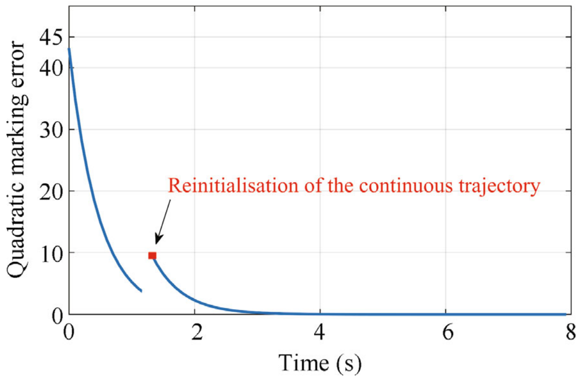

Fig. 7 Quadratic marking error of PN2

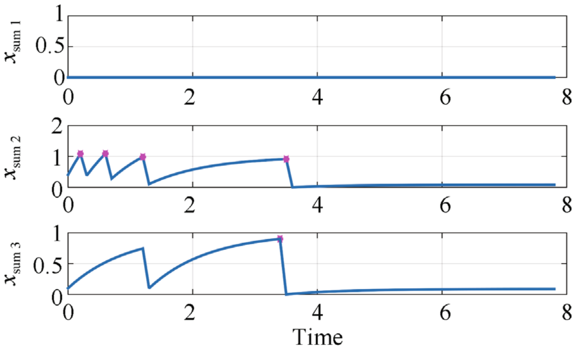

Fig. 8 Flow cumulative sum of PN2

Fig. 9 summarizes the obtained results. It displays the continuous and the discrete marking trajectories which are respectively represented by the solid and the dashed lines. In addition, it illustrates the timed control sequence σ^{timed} which is identified by the circle marks. The continuous trajectory is reinitialized once, which explains the discontinuity of the quadratic marking error J_M (Fig. 7).

The MPCC is applied to the TCPN and X_{sum} is computed at each sampling period. At $\tau_k = 0.2$ s, $x_{\text{sum } 2} \approx 1$, then t_2^c is the first selected continuous transition (see Fig. 8). The corresponding transition t_2^d is enabled at M_0 in the T-TDPN. Hence, $\sigma(1) = t_2^d$ and $M_{tot}^d(1) = \begin{bmatrix} 3 & 3 & 1 & 3 \end{bmatrix}^t$. Since t_2^d minimizes the criterion Φ $(t^{d\_opt} = t_2^d)$, $x_{\text{sum } 2} \leftarrow 0$. The same procedure is done for the step $h = 2$. For $h = 3$, t_2^c is also selected to fire (i.e., $x_{\text{sum } 2} \approx 1$) and it is enabled at $M_{tot}^d(2)$. However, this transition does not minimize Φ. Therefore, the continuous trajectory is reinitialized by the obtained discrete marking $M_{tot}^d(3) = \begin{bmatrix} 5 & 1 & 1 & 1 \end{bmatrix}^t$ that results from the firing of t_2^d (see Figs. 7 and 9), $\sigma(3) = t_2^d$, X_{sum} is reset to zero for all transitions (see Fig. 8) and the optimization problem is repeated. At the end of the simulation, the obtained control sequence is $\sigma = t_2^d\ t_2^d\ t_2^d\ t_3^d\ t_2^d$. The temporal control sequence, with respect to time constraints, is given by $\sigma^{timed} = (t_2^d, 1)(t_2^d, 1)(t_2^d, 1)(t_3^d, 1)(t_2^d, 1)$ such that:

$$\begin{bmatrix} 2 \\ 4 \\ 1 \\ 4 \end{bmatrix} \overset{(t_2^d,1)}{\longrightarrow} \begin{bmatrix} 3 \\ 3 \\ 1 \\ 3 \end{bmatrix} \overset{(t_2^d,1)}{\longrightarrow} \begin{bmatrix} 4 \\ 2 \\ 1 \\ 2 \end{bmatrix} \overset{(t_2^d,1)}{\longrightarrow} \begin{bmatrix} 5 \\ 1 \\ 1 \\ 1 \end{bmatrix} \overset{(t_3^d,1)}{\longrightarrow} \begin{bmatrix} 6 \\ 1 \\ 0 \\ 4 \end{bmatrix} \overset{(t_2^d,1)}{\longrightarrow} \begin{bmatrix} 7 \\ 0 \\ 0 \\ 3 \end{bmatrix}. \tag{42}$$

5 Conclusions

This article has proposed a new approach entitled MPCC with an aim of controlling the T-TDPN thanks to $\theta-$TCPN. This approach represents an extension of MPC under constant control actions. It has been developed, at first, in order to limit the computational complexity resulting from the use of a large prediction horizon with $\theta-$TCPN. As a consequence, the proposed method is suitable to control $\theta-$TCPN with hill-climbing phases that require at first to move away from the reference before reaching it. In addition, in order to ensure the closed-loop stability in the neighborhood of the desired marking, a terminal constraint has been added. Then, in a systematic way, the control sequence that drives a deadlock-free T-TDPN, from an initial marking to the desired one has been determined thanks to the algorithm driven by the control of the equivalent $\theta-$TCPN. Such an algorithm does not require the computation of the reachability graph. But, due to the local optimization, one should notice that the obtained control of the T-TDPN (if it exists) is not necessarily a global optimal control. In addition, no formal convergence proof has been provided since the proposed approach ensures convergence only when the conditions of Proposition 4 are satisfied. In our future work, the proposed approach will be extended to take into consideration the case when the selected transition to fire is not enabled at the discrete marking. Other decisions could be explored in order to improve Algorithm 3. For instance, since the T-TDPN is deadlock-free, there is always, at least, an enabled transition. Thereby, the control action will be selected among the enabled transitions such that it leads to the closest discrete marking from the continuous trajectory. The convergence of the T-TDPN will be investigated in that case. Moreover, an enhanced study on the conservation of some properties, like liveness and boundedness, of the T-TDPN via the fluidification and the MPCC will be carried out. Finally, the rejection of perturbations (i.e., the firing of unexpected transitions) will be studied in order to consider discrete control problems in uncertain functioning conditions.

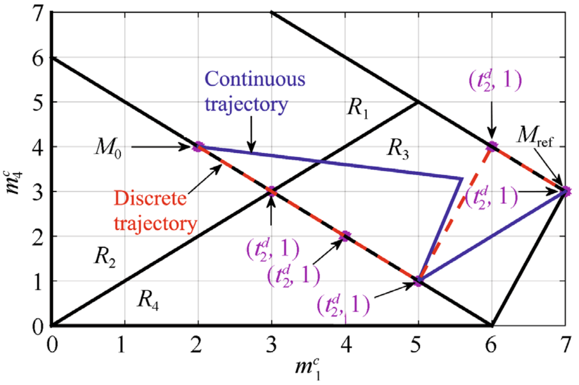

Fig. 9 Marking trajectories of the TCPN and the T-TDPN depicted in Fig. 6

References

[1] H. S. Hu, C. Chen, R. Su, Y. Liu, M. C. Zhou. Distributed supervisor synthesis for automated manufacturing systems using Petri nets. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Hong Kong, China, pp. 4423–4429, 2014.

[2] H. S. Hu, R. Su, M. C. Zhou, Y. Liu. Polynomially complex synthesis of distributed supervisors for large-scale AMSs using Petri nets. *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1–13.

[3] A. R. Moro, H. Yu, G. Kelleher. Hybrid heuristic search for the scheduling of flexible manufacturing systems using

Petri nets. *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 240–245, 2002.

[4] Y. Yang, H. S. Hu, Y. Liu. A petri net-based distributed control of automated manufacturing systems with assembly operations. In *Proceedings of IEEE International Conference on Automation Science and Engineering*, IEEE, Gothenburg, Sweden, pp. 1090–1097, 2015.

[5] H. Yu, A. Reyes, S. Cang, S. Lloyd. Combined Petri net modelling and AI based heuristic hybrid search for flexible manufacturing systems, Part I. Petri net modelling and heuristic search. *Computers & Industrial Engineering*, vol. 44, no. 4, pp. 527–543, 2003.

[6] H. Yu, A. Reyes, S. Cang, S. Lloyd. Combined Petri net modelling and AI-based heuristic hybrid search for flexible manufacturing systems, Part II. Heuristic hybrid search. *Computers & Industrial Engineering*, vol. 44, no. 4, pp. 545–566, 2003.

[7] B. Gudiño-Mendoza, E. López-Mellado. Modelling networked agents′ behaviour using timed hybrid Petri nets. *Procedia Technology*, vol. 7, pp. 289–296, 2013.

[8] N. Smata, D. Boudebous, J. Boukachour, S. Benmansour, C. Tolba. Production, supply, and traffic systems: A unified modelling using Petri nets. In *Proceedings of the 4th International Conference on Logistics*, IEEE, Hammamet, Tunisia, pp. 405–411, 2011.

[9] C. Tolba, D. Lefebvre, P. Thomas, A. El Moudni. Continuous and timed Petri nets for the macroscopic and microscopic traffic flow modelling. *Simulation Modeling Practice and Theory*, vol. 13, no. 5, pp. 407–436, 2005.

[10] D. Y. Lee, F. DiCesare. Scheduling flexible manufacturing systems using Petri nets and heuristic search. *IEEE Transactions on Robotics and Automation*, vol. 10, no. 2, pp. 123–132, 1994.

[11] Y. W. Kim, T. Suzuki, T. Narikiyo. FMS scheduling based on timed Petri net model and reactive graph search. *Applied Mathematical Modelling*, vol. 31, no. 6, pp. 955–970, 2007.

[12] E. Grolleau, A. Choquet-Geniet. Off-line computation of real-time schedules using Petri nets. *Discrete Event Dynamic Systems*, vol. 12, no. 3, pp. 311–333, 2002.

[13] D. Lefebvre, E. Leclercq. Control design for trajectory tracking with untimed Petri nets. *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1921–1926, 2015.

[14] R. David, H. Alla. Continuous Petri nets. In *Proceedings of the 8th European Workshop on Application and Theory of Petri Nets*, Zaragoza, Spain, pp. 275–294, 1987.

[15] M. Silva, E. Terue, J. M. Colom. Linear algebraic and linear programming techniques for the analysis of place/transition net systems. *Lectures on Petri Nets I: Basic Models, Lecture Notes in Computer Science*, Springer, Berlin Heidelberg, Germany vol. 1491, pp. 309–373, 1998.

[16] A. Giua, C. Mahulea, L. Recalde, C. Seatzu, M. Silva. Optimal control of continuous Petri nets via model predictive control. In *Proceedings of the 8th International Workshop on Discrete Event Systems*, IEEE, Ann Arbor, USA, pp. 235–241, 2006.

[17] M. Silva, L. Recalde. Petri nets and integrality relaxations: A view of continuous Petri net models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 4, pp. 314–327, 2002.

[18] M. Silva. Half a century after Carl Adam Petri′s Ph. D. thesis: A perspective on the field. *Annual Reviews in Control*, vol. 37, no. 2, pp. 191–219, 2013.

[19] C. R. Vásquez, M. Silva. Piecewise-linear constrained control for timed continuous Petri nets. In *Proceedings of the 48th IEEE Conference on Decision and Control, and the 2009 28th Chinese Control Conference*, IEEE, Shanghai, China, pp. 5714–5720, 2009.

[20] C. Mahulea, A. Ramirez-Trevino, L. Recalde, M. Silva. Steady-state control reference and token conservation laws in continuous Petri net systems. *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 2, pp. 307–320, 2008.

[21] H. Apaydin-Ózkan, J. Júlvez, C. Mahulea, M. Silva. Approaching minimum time control of timed continuous Petri nets. *Nonlinear Analysis: Hybrid Systems*, vol. 5, no. 2, pp. 136–148, 2011.

[22] R. Kara, M. Ahmane, J. J. Loiseau, S. Djennoune. Constrained regulation of continuous Petri nets. *Nonlinear Analysis: Hybrid Systems*, vol. 3, no. 4, pp. 738–748, 2009.

[23] E. Leclercq, D. Lefebvre. Feasibility of piecewise-constant control sequences for timed continuous Petri nets. *Automatica*, vol. 49, no. 12, pp. 3654–3660, 2013.

[24] D. Lefebvre, E. Leclercq, F. Druaux, P. Thomas. Gradient-based controllers for timed continuous Petri nets. *International Journal of Systems Science*, vol. 46, no. 9, pp. 1661–1678, 2015.

[25] J. Richalet, A. Rault, J. L. Testud, J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.

[26] J. J. Julvez, R. K. Boel. A continuous Petri net approach for model predictive control of traffic systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 40, no. 4, pp. 686–697, 2010.

[27] L. W. Wang, C. Mahulea, M. Silva. Distributed model predictive control of timed continuous Petri nets. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, IEEE, Firenze, Italy, pp. 6317–6322, 2013.

[28] M. Taleb, E. Leclercq, D. Lefebvre. Control design of timed Petri nets via model predictive control with ContPNs. In *Proceedings of the 12th International Workshop on Discrete Event Systems*, IFAC, Paris, France, pp. 149–154, 2014.

[29] M. Taleb, E. Leclercq, D. Lefebvre. Limitation of flow variation of timed continuous Petri nets via model predictive control. In *Proceedings of American Control Conference*, IEEE, Portland, Oregon, pp. 4919–4924, 2014.

[30] M. Taleb, E. Leclercq, D. Lefebvre. Limitation of flow variation of timed continuous Petri nets via model predictive control and Lyapunov criterion. In *Proceeding of European Control Conference*, IEEE, Strasbourg, France, pp. 1825–1830, 2014.

[31] R. David, H. Alla. *Petri Nets and Grafcet: Tools for Modelling Discrete Event Systems*, New York, USA: Prentice Hall, 1992.

[32] M. Silva. Introducing Petri nets. *Practice of Petri Nets in Manufacturing*, Springer, Netherlands, pp. 1–62, 1993.

[33] M. Silva, L. Recalde. On fluidification of Petri nets: From discrete to hybrid and continuous models. *Annual Reviews in Control*, vol. 28, no. 2, pp. 253–266, 2004.

[34] C. Mahulea, A. Giua, L. Recalde, C. Seatzu, M. Silva. Optimal model predictive control of timed continuous Petri nets. *IEEE Transactions on Automatic Control*, vol. 53, no. 7, pp. 1731–1735, 2008.

[35] M. Silva, L. Recalde. Continuization of timed Petri nets: From performance evaluation to observation and control. In *Proceedings of the 26th International Conference, Lecture Notes in Computer Science*, Springer, Miami, USA, vol. 3536, pp. 26–47, 2005.

[36] C. Mahulea, A. Giua, L. Recalde, C. Seatzu, M. Silva. On sampling continuous timed Petri Nets: Reachability "equivalence" under infinite servers semantics. In *Proceeding of the 2nd IFAC Conference on Analysis and Design of Hybrid Systems*, IFAC, Hotel Calabona, Italy, pp. 37–43, 2006.

[37] E. Jimenez, J. Julvez, L. Recalde, M. Silva. On controllability of timed continuous Petri net systems: The join free case. In *Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE, Seville, Spain, pp. 7645–7650, 2005.

[38] A. Geletu. Solving Optimization Problems Using the Matlab Optimization Toolbox – A Tutorial, 2007.

[39] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[40] J. Júlvez, L. Recalde, M. Silva. Steady-state performance evaluation of continuous mono-T-semiflow Petri nets. *Automatica*, vol. 41, no. 4, pp. 605–616, 2005.

Marwa Taleb received the M. Sc. degree in telecommunications from National Engineering School of Tunis, Tunisia in 2012. She received the Ph. D. degree in computer engineering, automation control and signal processing from University of Le Havre, France in 2016. She is actually temporarily attached to education and research with the Research Group on Electrical Engineering and Automatic Control.

Her research interests include dynamic systems, Petri nets and model predictive control.

E-mail: marwa.taleb@univ-lehavre.fr (Corresponding author)

ORCID iD: 0000-0002-5579-7569

Edouard Leclercq received the B. Sc. degree in physics and mathematics from Paris Educational District, France in 1987, the M. Sc. degree in electronics from University of Rouen, France in 1994, and the Ph. D. degree in automation from University of Le Havre, France in 1999. Since 1999 he is a lecturer at the Faculty of Sciences and Technology of Le Havre, France. Since 1999 he is with the G.R.E.A.H. (Electric and Automatic Engineering Research Group).

His research interests include modeling, control and fault detection, neural networks and Petri nets.

E-mail: edouard.leclercq@univ-lehavre.fr

Dimitri Lefebvre graduated from the Ecole Centrale of Lille, France in 1992. He received a Ph. D. degree in automatic control and computer science from University of Sciences and Technologies, Lille in 1994, and a HDR from University of Franche Comt Belfort, France in 2000. Since 2001, he has been a professor at Institute of Technology and Faculty of Sciences, University Le Havre, France. He is with the Research Group on Electrical Engineering and Automatic Control (GREAH) and from 2007 to 2012 he was the head of the group.

His research interests include Petri nets, learning processes, adaptive control, fault detection and diagnosis and its applications to electrical engineering

E-mail: dimitri.lefebvre@univ-lehavre.fr