ON/OFF strategy based minimum-time control of continuous Petri nets
-draft-

Liewei Wang, Cristian Mahulea, Jorge Júlvez and Manuel Silva*

December 23, 2014

Abstract

This paper focuses on the target marking control problem of *timed continuous Petri nets* (TCPN), aiming to drive the system from an initial state to a desired final one. This problem is similar to the set-point control problem in a general continuous-state system. In a previous work, a simple and efficient ON/OFF controller was proposed for Choice-Free nets, and it was proved to be minimum-time [31]. However, for general TCPN the ON/OFF controller may bring the system to "blocking" situations due to its "greedy" firing strategy, and the convergence to the final state is not ensured. In this work the ON/OFF controller is extended to general TCPN by adding more "fair" strategies to solve conflicts in the system: the ON/OFF+ controller is obtained by forcing proportional firings of conflicting transitions. Nevertheless, such kind of controller might highly slow down the system when transitions have flows of different orders of magnitude, therefore a balancing process is introduced, leading to the B-ON/OFF controller. A third approach introduced here is the MPC-ON/OFF controller, a combination of Model Predictive Control (MPC) and the ON/OFF strategy; it may achieve a smaller number of time steps for reaching the final states, but usually requires more CPU time for computing the control laws. All the proposed extensions are heuristic methods for the minimum-time control and their convergences are proved. Finally, an application example of a manufacturing cell is considered to illustrate the methods. It is shown that by using the proposed controllers, reasonable numbers of time steps for reaching the final state can be obtained with low computational complexity.

Published as:
L. Wang, C. Mahulea, J. Julvez, and M. Silva, "ON/OFF strategy based minimum-time control of continuous Petri nets," Nonlinear Analysis: Hybrid Systems, vol. 12, pp. 50-65, May 2014. DOI: 10.1016/j.nahs.2013.11.007

*The authors are with the Aragón Institute of Engineering Research (I3A), University of Zaragoza, Maria de Luna 1, 50018, Spain.
This work has been partially supported by CICYT - FEDER grant DPI2010-20413

1 Introduction

Petri Nets (PN) constitute a well known paradigm used for modeling, analysis, and synthesis of *discrete event systems* (DES). Straightforwardly depicting sequences, concurrency, conflicts and synchronizations, it is widely applied in the industry for the analysis of manufacturing, traffic, or software systems, for example [11, 12, 17]. Similarly to other modeling formalisms for DES, it also suffers from the *state explosion* problem. To overcome it, a classical relaxation technique called *fluidization* can be used.

Continuous PN (CPN) [4, 24] are fluid approximations of classical *discrete PN* obtained by removing the integrality constraints, which means that the firing count vectors (and consequently the markings) are no longer restricted to be in the natural numbers but relaxed into the non-negative real numbers. An important advantage of this relaxation is that more efficient algorithms are available for their analysis. A simple and interesting way to introduce time to CPN is to assume that time is associated to transitions, obtaining timed CPN (TCPN).

One of the important objectives in the control of CPN is to drive the system from an initial state $\boldsymbol{m}_0$ to a desired final state $\boldsymbol{m}_f$, which is similar to the *set-point* control problem in a general continuous-state system. This desired state can be selected, in a preliminarily planning stage [25], according to some optimality criteria, e.g., maximizing the flows. By considering the CPN as a relaxation of discrete systems, the continuous state can be viewed as an approximation of the average state in its original discrete system. Several approaches can be found in the literature for handling this control problem assuming that all transitions are controllable, for example, in [5, 13, 21, 6, 8, 29]. In [27], the control methods are first carried out in the continuous model, after that they are applied back to control the original discrete one. Many of these works focus on *infinite server semantics*, which has been proved to provide a better approximation of discrete systems for a broad class of nets [15], also most probably in general. In the case of systems with uncontrollable transitions, the control problem may become much more complex [10, 28].

The time spent for reaching the final state, and the corresponding complexity for computing the control laws, are naturally important in the control synthesis process. Minimum-time control is a classical class of problem that has been widely studied (see, for example [3, 9, 23, 2]). Nevertheless, several peculiarities of TCPN motivate the search for new approaches: (1) the existence of a meaningful net structure, from which many behavioral properties can be derived for a given initial marking; (2) control actions are asymmetrically bounded and the upper bounds depend on the state (the marking); (3) the existence of *minimum* operator under infinite sever semantics. Some a few works can be found in the literature dealing with this topic; for instance, [1] proposed a heuristic for minimum-time control of TCPN that drives the system to the final state by following a piecewise-straight marking trajectory.

In this work, we will consider the ON/OFF (also known as *Bang-Bang*) controller, a feedback controller that switches from one extreme to the other at certain times (switching points). ON/OFF strategies frequently arise in minimum-time problems and many classical results related to the minimum-time control of linear systems exist (see, for example, the previously cited works, also [26, 16]). However, we may not be able to apply them directly to our case because, as has been already mentioned, TCPN under infinite server semantics is a piecewise linear system with non-negative states and dynamically bounded control variables.

An ON/OFF controller has been proposed for Choice-Free (CF) (or structurally persistent) net systems in [31]. It is efficient because the minimum-time state evolution is ensured; it also has very low computational complexity: only a *minimal firing count vector* needs to be computed, and a simple linear programming problem (LPP) is solved at each time step. The essential problem of this standard ON/OFF controller is that when there are conflicts, this “greedy” strategy of firing transitions may bring the system to a “blocking” situation (see Ex. 3.1 for an example).

In this work, the ON/OFF control scheme is further investigated and three heuristic extensions are presented for general net systems, ensuring that the final state is reached in finite time. Even if in this case the minimum-time is not guaranteed, reasonable numbers of time steps are obtained. By forcing the conflicting transitions to fire proportionally to the given firing count vector, we obtain the ON/OFF+ controller. With very low computational complexity, nevertheless a possible drawback of this method is the following: the firing speeds of transitions in a conflict are determined by the "slower" ones, and the overall system may be highly slowed down in some extreme cases. Therefore, the second extension, *balanced* ON/OFF (B-ON/OFF) controller is proposed; it tries to balance the "fast" and "slow" conflicting transitions before applying the pure ON/OFF+ controller. These two extensions have been initially discussed in a previous work [30]; in this paper a third method is considered. It is a combination of Model Predictive Control (MPC) and ON/OFF strategy: solving the conflicts using MPC and firing other transitions using the ON/OFF strategy; the asymptotic stability for this MPC-based approach is also proved. The first two methods have very low computational complexity, while by using the MPC-ON/OFF controller we may reach the final state faster (using larger time horizon), but with higher computational complexity. We want to stress that all the extensions presented here are only heuristics for minimum-time control, and the optimal solution for the minimum-time control is difficult to obtain when dealing with general net system structures, because of the non-linearity of the system dynamics and of the state (marking) dependent constraints for the control variables.

This paper is organized as follows: Section 2 briefly recalls some basic concepts of CPN. In Section 3 the standard ON/OFF controller is recalled and its main drawback is stated. Three ON/OFF strategy based extensions are proposed in Section 4. Section 5 illustrates a manufacturing cell case study. Conclusions are given in section 6.

2 Basic Concepts and Notations

2.1 Continuous Petri Nets

The reader is assumed to be familiar with the basic concepts of continuous Petri nets (see [4, 24] for a gentle introduction).

Definition 2.1. *A continuous PN system is a pair* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ *where* $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$ *is a net structure where:*

- P *and* T *are the sets of places and transitions respectively.*
- $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{N}_{\geq 0}^{|P| \times |T|}$ *are the pre and post incidence matrices.*
- $\boldsymbol{m}_0 \in \mathbb{R}_{\geq 0}^{|P|}$ *is the initial marking (state).*

Let p_i, $i = 1, \ldots, |P|$ and t_j, $j = 1, \ldots, |T|$ denote the places and transitions. In the incidence matrices, $\boldsymbol{Pre}[p_i, t_j] = w_1$ and $\boldsymbol{Post}[p_i, t_j] = w_2$ indicate the connections between places and transitions: if $w_1 > 0$ there is an arc from p_i to t_j with w_1 as the weight; if $w_2 > 0$ there is an arc from t_j to p_2 with w_2 as the weight. For $v \in P \cup T$, the sets of its input and output nodes are denoted as ${}^\bullet v$ and $v^\bullet$, respectively. These definitions can be naturally extended to a set of nodes. Each place can contain a non-negative real number of tokens, its marking. The distribution of tokens in places is denoted by $\boldsymbol{m}$ and $\boldsymbol{m}[p_i]$ represents the marking of place p_i. The enabling degree of a transition $t_j \in T$ is given by:

$$enab(t_j, \boldsymbol{m}) = \min_{p_i \in {}^\bullet t_j} \left\{ \frac{\boldsymbol{m}[p_i]}{\boldsymbol{Pre}[p_i, t_j]} \right\}$$

which represents the maximum amount in which t_j can fire. Transition t_j is called *k-enabled* at marking $\boldsymbol{m}$, if $enab(t_j, \boldsymbol{m}) = k$, being enabled if $k > 0$. An enabled transition t_j can fire in any real amount α, with $0 < \alpha \leq enab(t_j, \boldsymbol{m})$ leading to a new state $\boldsymbol{m'} = \boldsymbol{m} + \alpha \cdot \boldsymbol{C}[\cdot, t_j]$ where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the *token flow matrix* and $\boldsymbol{C}[\cdot, j]$ is its j^{th} column.

Non negative left and right natural annullers of the token flow matrix $\boldsymbol{C}$ are called P-*semiflows*, denoted by $\boldsymbol{y}$, and T-semiflows, denoted by $\boldsymbol{x}$, respectively. If $\exists \boldsymbol{y} > 0$, $\boldsymbol{y} \cdot \boldsymbol{C} = 0$, then the net is said to be conservative. If $\exists \boldsymbol{x} > 0$, $\boldsymbol{C} \cdot \boldsymbol{x} = 0$ it is said to be consistent.

A PN system is bounded when every place is bounded, i.e., its token content is less than some bounds at every reachable marking. It is live when every transition is live, i.e., it can ultimately fire from every reachable marking. If for all $p \in P$, $|p^\bullet| \leq 1$ then $\mathcal{N}$ is called *Choice-Free* (CF). A CF net is *structurally persistent* in the sense that independently of the initial marking, the net has no conflict. A net $\mathcal{N}$ is *Equal-Conflict* (EQ) if for any two transition t_1, t_2, ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset \Rightarrow \boldsymbol{Pre}[\cdot, t_1] = \boldsymbol{Pre}[\cdot, t_2]$.

If $\boldsymbol{m}$ is reachable from $\boldsymbol{m}_0$ through a finite sequence σ, the state (or fundamental) equation is satisfied: $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, where $\boldsymbol{\sigma} \in \mathbb{R}^{|T|}_{\geq 0}$ is the *firing count vector*, i.e., $\boldsymbol{\sigma}[t_j]$ is the cumulative amount of firings of t_j in the sequence σ. The *reachability space* of a given system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$, denoted by $\text{RS}(\mathcal{N}, \boldsymbol{m}_0)$, is the set of all markings that are reachable by a finite firing sequence. If the net system is consistent and every transition can be fired at least once, i.e., there exists no empty *siphon* at $\boldsymbol{m}_0$, then any solution $\boldsymbol{m} > 0$ of the state equation is reachable ([19]). A firing count vector $\boldsymbol{\sigma}$ is said to be *minimal* if for any T-*semiflow* $\boldsymbol{x}$, $||\boldsymbol{x}|| \nsubseteq ||\boldsymbol{\sigma}||$, where $||\cdot||$ stands for the support of a vector, i.e., the index of the elements different than zero. One way to compute a minimal firing count vector $\boldsymbol{\sigma}$ that drives the system from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$ is by solving the LPP:

$$\begin{array}{ll} min & \mathbf{1}^T \cdot \boldsymbol{\sigma} \\ \text{s.t.} & \boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma} \\ & \boldsymbol{\sigma} \geq 0 \end{array} \tag{1}$$

In timed continuous PN (TCPN) the state equation has an explicit dependence on time: $\boldsymbol{m}(\tau) = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}(\tau)$, which through time differentiation becomes $\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \dot{\boldsymbol{\sigma}}(\tau)$. The derivative of the firing count $\boldsymbol{f}(\tau) = \dot{\boldsymbol{\sigma}}(\tau)$ is called the *firing flow*. For the sake of clarity, τ will be omitted in the rest of the paper when there is no confusion.

Depending on how the flow is defined, many firing server semantics appear. The *infinite* (or variable speed) and the *finite* (or constant speed) server semantics ([4, 24]) are the most used ones, where a firing rate $\lambda_j \in \mathbb{R}_{>0}$, or denoted by $\boldsymbol{\lambda}[t_j]$, is associated to each transition t_j. This work deals with *infinite server semantics*, for which the flow of a transition t_j at marking $\boldsymbol{m}$ is the product of its firing rate and its enabling degree:

$$\boldsymbol{f}[t_j] = \lambda_j \cdot enab(t_j, \boldsymbol{m}) = \lambda_j \cdot \min_{p_i \in {}^\bullet t_j} \left\{ \frac{\boldsymbol{m}[p_i]}{\boldsymbol{Pre}[p_i, t_j]} \right\} \tag{2}$$

Due to the existence of *minimum* operator, the dynamical system corresponds to a *piecewise linear* system and it induces two strongly related concepts: a) the set of places defining the enabling degree of transitions is known as *configuration*; b) the sub-state space, in which the configuration is identical, is known as *region*. More formally:

Definition 2.2. *A* configuration *of a net $\mathcal{N}$ is a set of (p, t) arcs, one per transition,* covering *the set T of transitions. Associated to a given configuration $\mathcal{C}_k$ is the following $|T| \times |P|$ configuration matrix:*

$$\boldsymbol{\Pi}_k[t, p] = \begin{cases} \frac{1}{\boldsymbol{Pre}[p,t]}, & \textit{if } (p, t) \in \mathcal{C}_k \\ 0, & \textit{otherwise} \end{cases} \tag{3}$$

In the case of a TCPN system under infinite server semantics, at a given marking $\boldsymbol{m} \in \text{RS}(\mathcal{N}, \boldsymbol{m}_0)$ the flow of a transition t_j, given by (2), is defined by the marking of an input place $p_i \in {}^\bullet t_j$, the one that gives the minimum. Let us notice that the reachability set $\text{RS}(\mathcal{N}, \boldsymbol{m}_0)$ of a TCPN system can be partitioned (except on the borders) according to the configurations and inside each obtained *convex region* $\mathcal{R}_i(\mathcal{N}, \boldsymbol{m}_0)$ the system dynamic is linear.

2.2 Control Problem

In this paper we consider the net system to be subject to external control actions, and we assume that the only admissible control law consists in *slowing down* the (maximal) firing flow of transitions (defined for the *uncontrolled* systems) [24]. This means that transitions modeling machines, for example, cannot work faster than their nominal speeds. Under this assumption, the controlled flow of a TCPN system is denoted as:

$$\boldsymbol{w}(\tau) = \boldsymbol{f}(\tau) - \boldsymbol{u}(\tau), \text{with } 0 \leq \boldsymbol{u}(\tau) \leq \boldsymbol{f}(\tau)$$

The overall behavior of the system is ruled by:

$$\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot (\boldsymbol{f}(\tau) - \boldsymbol{u}(\tau))$$

It is assumed that every transition is *controllable* (t_j is uncontrollable if no control can be applied to it, i.e, $\boldsymbol{u}[t_j] = 0$). The control problem addressed here is to design a control action $\boldsymbol{u}$ that drives the system from an initial marking $\boldsymbol{m}_0$ to a desired final marking $\boldsymbol{m}_f$, trying to minimize the time spent on the trajectory.

In the sequel, we usually assume $\boldsymbol{m}_0 > 0$. This assumption is rather weak, since any practical system should fire any transition. Moreover, it should be noticed that for TCPN under infinite server semantics, once a place is marked it will take infinite time to be emptied (like the theoretical discharging of a capacitor in an electrical RC-circuit). Therefore, if there exist places that must be emptied during the trajectory to $\boldsymbol{m}_f$, the final marking can only be reached at the limit, i.e., in infinite time. Thus we also assume $\boldsymbol{m}_f > 0$, i.e., an interior point.

3 ON/OFF controller and its drawbacks

By sampling the continuous-time CPN system with a *sampling period* Θ, we obtain the discrete-time CPN ([13]) given by:

$$\begin{aligned} &\boldsymbol{m}_{k+1} = \boldsymbol{m}_k + \boldsymbol{C} \cdot \boldsymbol{w}_k \cdot \Theta \\ &0 \leq \boldsymbol{w}_k \leq \boldsymbol{f}_k \end{aligned} \tag{4}$$

Here $\boldsymbol{m}_k$, $\boldsymbol{w}_k$ and $\boldsymbol{f}_k$ are the marking, controlled flow and uncontrolled flow at sampling instant k, i.e., at $\tau = k \cdot \Theta$.

It is proved in [13] that if the sampling period satisfies (5), the reachability spaces of discrete-time and continuous-time CPN systems are the same, excepting at borders. In this paper, we assume that (5) is satisfied.

$$\forall p \in P : \sum_{t_j \in p^\bullet} \lambda_j \cdot \Theta < 1 \tag{5}$$

An ON/OFF controller is presented in [31] for CF nets, where every transition fires as fast as possible at any time step until an upper bound, the minimal firing count vector $\boldsymbol{\sigma}$, is reached. It is first proposed based on the discrete-time CPN model, then extended to continuous-time CPN. Algorithm 1 synthesizes the ON/OFF controller.

The main advantage of the ON/OFF controller is its low computational complexity. Given a (minimal) firing count vector (which can be computed in polynomial time), the control actions are obtained by solving a simple LPP at each time step. It is proved that in the case of CF nets, using the minimum firing count vector, the ON/OFF controller drives the system to its final state in minimum-time. Nevertheless, in the case of general nets the convergence to the final state may not be ensured—in general non-CF nets, conflicts ($|p^\bullet| > 1$) may appear, thus firing faster one transition may reduce the firing of another transition and the overall time for reaching $\boldsymbol{m}_f$ may increase, being infinity in the extreme case. The following example shows a live and bounded net system, in which by applying the ON/OFF strategy the final state cannot be reached.

Algorithm 1 ON/OFF controller

Input: $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$, $\boldsymbol{m}_f$, $\boldsymbol{\sigma}$, Θ
Output: $\boldsymbol{w}_0$, $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, ...

$k \leftarrow 0$
while $\sum_{i=0}^{k-1} \boldsymbol{w}_i \cdot \Theta \neq \boldsymbol{\sigma}$ **do**
    Solve the following LPP:

$$
\begin{array}{ll}
max & \mathbf{1}^T \cdot \boldsymbol{w}_k \\
\text{s.t.} & \boldsymbol{m}_{k+1} = \boldsymbol{m}_k + \boldsymbol{C} \cdot \boldsymbol{w}_k \cdot \Theta \\
 & \mathbf{0} \leq \boldsymbol{w}_k \cdot \Theta \leq \boldsymbol{\sigma} - \sum_{i=0}^{k-1} \boldsymbol{w}_i \cdot \Theta \\
 & \boldsymbol{w}_k[t_j] \leq \lambda_j \cdot enab(t_j, \boldsymbol{m}_k), \forall t_j \in T \\
 & \boldsymbol{m}_{k+1} \geq 0
\end{array}
\tag{6}
$$

    Apply $\boldsymbol{w}_k : \boldsymbol{m}_{k+1} \leftarrow \boldsymbol{m}_k + \boldsymbol{C} \cdot \boldsymbol{w}_k \cdot \Theta$
    $k \leftarrow k + 1$
end while
return $\boldsymbol{w}_0$, $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, ...

Example 3.1. *Assume we want to drive the system in Fig.1 to final state* $\boldsymbol{m}_f = [0.5\ 0.5\ 0.5\ 0.5\ 0.2\ 0.4\ 1.4]^T$*, the firing rate of* t_3 *is 10, while the firing rates of other transitions are all set to 1.* $\boldsymbol{\sigma} = [0.8\ 1.3\ 0.5\ 0\ 1\ 0\ 0]^T$ *is a minimal firing count vector driving the system from* $\boldsymbol{m}_0$ *(shown in the figure) to* $\boldsymbol{m}_f$*. By using this setting and applying the ON/OFF controller,* $\boldsymbol{m}_f$ *cannot be reached and the system will be "blocked" in an intermediate marking* $\boldsymbol{m} = [1\ 0\ 0.78\ 0.22\ 0\ 0\ 2]^T$*. Notice that, this "blocking" situation is imposed by the controller. For instance, transition* t_7 *is actually enabled at* $\boldsymbol{m}$*, but the control law has forbidden its firing because* $\boldsymbol{\sigma}[t_7] = 0$*.*

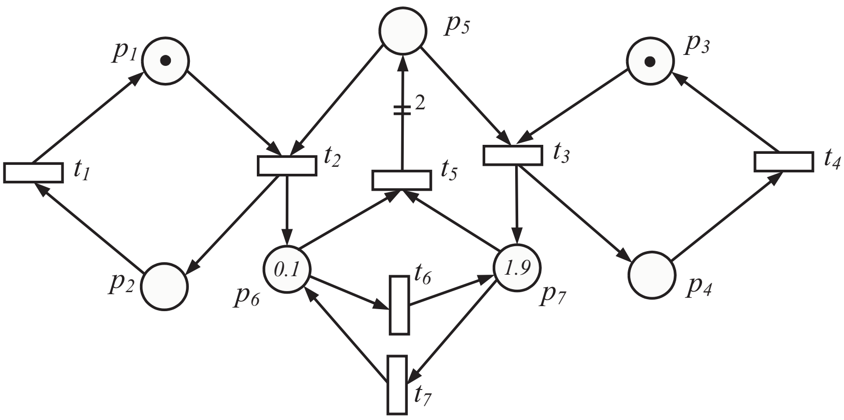


Figure 1: A live and bounded CPN system that the ON/OFF controller brings to a "blocking" situation

One may think that deadlock-freeness is a sufficient condition for applying the ON/OFF controller to a general net system. But we should notice that, the control laws may forbid the firings of some transitions (like in Ex.3.1, the firing of t_7 is forbidden because $\boldsymbol{\sigma}[t_7] = 0$), bringing the system to "blocking" situations. Therefore, in order to apply the ON/OFF controller we may need a tedious process after each time step to check if the system has been "blocked" and change to other control laws when it occurs. For instance, in Ex.3.1, after the system has been "blocked" in $\boldsymbol{m}$ we can change to another control law $\boldsymbol{\sigma'} = [0\ 0.5\ 0.28\ 0\ 0.49\ 0\ 0.39]^T$ and continue to apply the ON/OFF strategy, then the system could converge to $\boldsymbol{m}_f$.

4 Extended ON/OFF based methods

Because the ON/OFF controller cannot be directly applied to general TCPN, three heuristic extensions are proposed: ON/OFF+, B-ON/OFF and MPC-ON/OFF. In all the methods the convergence to the final state is always guaranteed, although we may not obtain a minimum-time state evolution. The ON/OFF+ overcomes the problem of the standard ON/OFF controller by forcing proportional firings of conflicting transitions; B-ON/OFF is proposed to handle those bad cases of applying the ON/OFF+ controller; the MPC-ON/OFF controller has higher computational complexity, but may need less time for reaching the final state.

4.1 ON/OFF+ controller

The problem of the ON/OFF controller arises from "an inappropriate" manner of solving the conflicts (e.g., in the system of Fig.1, since $\lambda_3 \gg \lambda_2$, t_3 fires much faster than t_2). Two transitions t_a and t_b are in a structural conflict relation if ${}^\bullet t_a \cap {}^\bullet t_b \neq \emptyset$. Here, let us define the *coupled conflict* relation as its transitive closure.

In order to overcome this problem, we consider a more "fair" strategy to solve the conflicts: forcing the flows of transitions that are in coupled conflict relation to be proportional to the given firing count vector. Meanwhile, for the rest of (persistent) transitions the ON/OFF strategy is applied.

The modified ON/OFF controller is shown in Algorithm 2 and we will call it ON/OFF+ controller.

Algorithm 2 ON/OFF+ controller

Input: $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$, $\boldsymbol{m}_f$, $\boldsymbol{\sigma}$, Θ

Output: $\boldsymbol{w}_0$, $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, ...

1: $k \leftarrow 0$

2: **while** $\sum_{i=0}^{k-1} \boldsymbol{w}_i \cdot \Theta \neq \boldsymbol{\sigma}$ **do**

3: Solve the following LPP:

$$\begin{array}{ll} max & \mathbf{1}^T \cdot \boldsymbol{w}_k \\ \text{s.t.} & \boldsymbol{m}_{k+1} = \boldsymbol{m}_k + \boldsymbol{C} \cdot \boldsymbol{w}_k \cdot \Theta \\ & \mathbf{0} \leq \boldsymbol{w}_k \cdot \Theta \leq \boldsymbol{\sigma} - \sum_{i=0}^{k-1} \boldsymbol{w}_i \cdot \Theta \\ & \boldsymbol{w}_k[t_j] \leq \lambda_j \cdot enab(t_j, \boldsymbol{m}_k), \forall t_j \in T \\ & \boldsymbol{m}_{k+1} \geq 0 \\ & \boldsymbol{w}_k[t_a] \cdot \boldsymbol{\sigma}[t_b] = \boldsymbol{w}_k[t_b] \cdot \boldsymbol{\sigma}[t_a] \\ & \qquad \forall t_a, t_b, {}^\bullet t_a \cap {}^\bullet t_b \neq \emptyset \text{ and } \boldsymbol{\sigma}[t_a] > 0, \boldsymbol{\sigma}[t_b] > 0 \end{array} \tag{7}$$

4: Apply $\boldsymbol{w}_k : \boldsymbol{m}_{k+1} \leftarrow \boldsymbol{m}_k + \boldsymbol{C} \cdot \boldsymbol{w}_k \cdot \Theta$

5: $k \leftarrow k + 1$

6: **end while**

7: **return** $\boldsymbol{w}_0$, $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, ...

The procedure of the ON/OFF+ controller is similar to the one of the standard ON/OFF, except the last constraint of LPP (7) in the step 3 of Algorithm 2, which means that, at any time step k, if transitions t_a and t_b are in conflict, the following will be forced: $\frac{\boldsymbol{w}_k[t_a]}{\boldsymbol{w}_k[t_b]} = \frac{\boldsymbol{\sigma}[t_a]}{\boldsymbol{\sigma}[t_b]}$. Notice that, only transitions with positive values in the corresponding firing count vector should be considered. In the following, it is assumed that $\boldsymbol{m}_0 > 0$.

In order to prove the convergence, we first show that by using some reduction rules, the original system with the ON/OFF+ controller is equivalent to a CF net system with a particular controller $\mathcal{A}$, i.e., the same state trajectory can be obtained. Then, we prove that controller $\mathcal{A}$ drives the CF net system to $\boldsymbol{m}_f$, implying that the ON/OFF+ controller also drives the original one to $\boldsymbol{m}_f$.

Reduction Rule. *Given a net $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}\rangle$, let $T_j = \{t_1, t_2, ..., t_n\} \subseteq T$ be a set of transitions that are in coupled conflict relation. These transitions fire proportionally according to a given firing count vector $\boldsymbol{\sigma}$, i.e., for any $t_a, t_b \in T_j$, $\boldsymbol{\sigma}[t_a], \boldsymbol{\sigma}[t_b] > 0$, if t_a fires in an amount s_a, simultaneously, t_b fires in an amount s_b, such that $\frac{s_a}{s_b} = \frac{\boldsymbol{\sigma}[t_a]}{\boldsymbol{\sigma}[t_b]}$. Let $\bar{\sigma} = \sum_{t \in T_j} \boldsymbol{\sigma}[t]$, $\mathcal{N}$ is transformed to $\mathcal{N}' = \langle P, T', \boldsymbol{Pre}', \boldsymbol{Post}'\rangle$ in the following way:*

(1) $T' = T \setminus T_j$

(2) Merge T_j to a new transition t_j, $T' = T' \cup \{t_j\}$

(3) $\forall p \in {}^\bullet T_j$, $\boldsymbol{Pre}'[p, t_j] = \sum_{t \in p^\bullet} \boldsymbol{Pre}[p, t] \cdot \boldsymbol{\sigma}[t] / \bar{\sigma}$

(4) $\forall p \in T_j{}^\bullet$, $\boldsymbol{Post}'[p, t_j] = \sum_{t \in {}^\bullet p} \boldsymbol{Post}[p, t] \cdot \boldsymbol{\sigma}[t] / \bar{\sigma}$

Example 4.1. *Let $\boldsymbol{m} > 0$ and $\boldsymbol{\sigma}[t_1] > 0$, $\boldsymbol{\sigma}[t_2] > 0$. Fig. 2 shows how to merge two conflicting transitions t_1 and t_2 to t_{1_2}.*

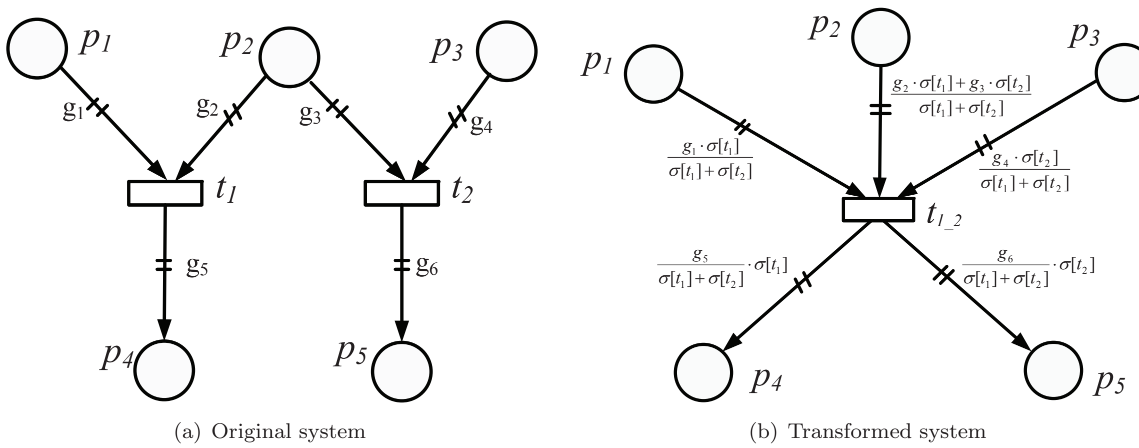


Figure 2: Reduction rule: merging t_1 and t_2

Proposition 4.2. *Let $\mathcal{S} = \langle \mathcal{N}, \boldsymbol{m}_0\rangle$, and $\mathcal{S}' = \langle \mathcal{N}', \boldsymbol{m}_0\rangle$ be the transformed system from $\mathcal{S}$ by merging $T_j = \{t_1, t_2, ..., t_n\}$ to t_j by using the reduction rule. If in $\mathcal{S}$, the transitions in T_j fire proportionally according to a given firing count vector $\boldsymbol{\sigma}$, and in $\mathcal{S}'$, transition t_j fires in an amount equal to the sum of the firing amounts of transitions in T_j, then the same marking is reached in $\mathcal{S}$ and $\mathcal{S}'$.*

Proof: It follows immediately by the definition of the reduction rule. ■

For example, consider place p_2 in Fig.2, and let $s_1 = \alpha \cdot \boldsymbol{\sigma}[t_1]$, $s_2 = \alpha \cdot \boldsymbol{\sigma}[t_2]$, $\alpha > 0$. If $t_1(s_1)t_2(s_2)$ is fired in the original system, the new marking of p_2 is:

$$\boldsymbol{m}_1[p_2] = \boldsymbol{m}_0[p_2] - g_2 \cdot \alpha \cdot \boldsymbol{\sigma}[t_1] - g_3 \cdot \alpha \cdot \boldsymbol{\sigma}[t_2]$$

In the transformed system, if $t_{1\_2}(s_1 + s_2)$ is fired, the new making of p_2 is:

$$\begin{aligned}\boldsymbol{m}'_1[p_2] &= \boldsymbol{m}_0[p_2] - (s_1 + s_2) \cdot \frac{g_2 \cdot \boldsymbol{\sigma}[t_1] + g_3 \cdot \boldsymbol{\sigma}[t_2]}{\boldsymbol{\sigma}[t_1] + \boldsymbol{\sigma}[t_2]} \\ &= \boldsymbol{m}_0[p_2] - \alpha \cdot (\boldsymbol{\sigma}[t_1] + \boldsymbol{\sigma}[t_2]) \cdot \frac{g_2 \cdot \boldsymbol{\sigma}[t_1] + g_3 \cdot \boldsymbol{\sigma}[t_2]}{\boldsymbol{\sigma}[t_1] + \boldsymbol{\sigma}[t_2]} \\ &= \boldsymbol{m}_1[p_2].\end{aligned}$$

Similarly, markings of places p_1 and p_3 are also equal in both systems.

Corollary 4.3. *If $\boldsymbol{m}_f > 0$ is reachable in $\mathcal{S}$ by firing $\boldsymbol{\sigma}$ from $\boldsymbol{m}_0 > 0$, then $\boldsymbol{m}_f$ is reachable in $\mathcal{S}'$ by firing $\boldsymbol{\sigma}'$, where:*

$$\boldsymbol{\sigma}'[t_j] = \begin{cases} \sum_{t \in T_j} \boldsymbol{\sigma}[t] & \textit{if } t_j \textit{ is obtained by merging a set of transitions } T_j \\ \boldsymbol{\sigma}[t_j] & \textit{otherwise} \end{cases}$$

Proposition 4.4. *Let $\mathcal{S} = \langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a discrete-time TCPN system with $\boldsymbol{m}_0 > 0$ and Θ the sampling period. Let $\boldsymbol{m}_f > 0$ be a reachable final marking, such that $\boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$. By applying the ON/OFF+ controller, the system state converges to $\boldsymbol{m}_f$ in finite time.*

Proof: Let $\mathcal{S}' = \langle \mathcal{N}', \boldsymbol{\lambda}', \boldsymbol{m}_0 \rangle$ be the system transformed from $\mathcal{S}$ by merging all the conflicting transitions, using the reduction rule (therefore $\mathcal{S}'$ is CF).

Assume there exists a controller $\mathcal{A}$ applied to $\mathcal{S}'$, with $\boldsymbol{w}'_k[t_j]$ the controlled flow at each time step k, such that: (1) if t_j is obtained by merging a set of transitions T_j in a coupled conflict relation, we have $\boldsymbol{w}'_k[t_j] = \sum_{t \in T_j} \boldsymbol{w}_k[t]$; (2) otherwise $\boldsymbol{w}'_k[t_j] = \boldsymbol{w}_k[t_j]$, where $\boldsymbol{w}_k[t_j]$ is the flow of transition t_j in $\mathcal{S}$ that is controlled by using the ON/OFF+ controller. Then, according to Proposition 4.2, the state trajectory of $\mathcal{S}'$ obtained by applying controller $\mathcal{A}$ is the same as in $\mathcal{S}$ obtained by applying the ON/OFF+ controller. Therefore it is equivalent to prove that by applying controller $\mathcal{A}$ to $\mathcal{S}'$, $\boldsymbol{m}_f$ is reached in finite time.

This controller $\mathcal{A}$ always exists, because if the firing rate of t_j in $\mathcal{S}'$, λ'_j, is large enough, case (1) can always be satisfied, by using a positive control action $\boldsymbol{u}_k[t_j]$. In particular, it is defined as:

$$\boldsymbol{u}_k[t_j] = \lambda'_j \cdot enab(t_j, \boldsymbol{m}_k) - x_k^j \tag{8}$$

where x_k^j is obtained by solving the LPP (9):

$$\begin{aligned} x_k^j = \quad & max \quad \sum_{t_d \in T_j} x_k^d \\ & \text{s.t.} \quad x_k^a \cdot \boldsymbol{\sigma}[t_b] = x_k^b \cdot \boldsymbol{\sigma}[t_a], \forall t_a, t_b \in T_j \\ & \qquad\; 0 \leq x_k^d \leq \lambda_d \cdot enab(t_d, \boldsymbol{m}_k), \forall t_d \in T_j \\ & \qquad\; \sum_{i=0}^{k} x_k^d \cdot \Theta \leq \boldsymbol{\sigma}[t_d], \forall t_d \in T_j \end{aligned} \tag{9}$$

where $enab(t_d, \boldsymbol{m}_k)$, $t_d \in T_j$, is the enabling degree of t_d in the original system at $\boldsymbol{m}_k$.

For case (2) we simply use the ON/OFF strategy and the same firing rate as in $\mathcal{S}$.

Finally, let us notice that $\mathcal{S}'$ is a CF net, so for sure controller $\mathcal{A}$ can drive $\mathcal{S}'$ to its final state in finite time [31], implying that by applying the ON/OFF+ controller to $\mathcal{S}$, the final state is also reached in finite time. ■

Given a firing count vector $\boldsymbol{\sigma}$, if transition t_j is a persistent one (transitions that are not in a conflict relation) and the goal is to minimize the time spent for firing $\boldsymbol{\sigma}[t_j]$, the ON/OFF strategy is optimal. For the transitions in conflict, the ON/OFF+ controller gives a way to handle their firings, but in general, it is just a heuristic method for the minimum-time.

Example 4.5. *Let us consider the net system in Fig. 3. Assume that the desired final state is $\boldsymbol{m}_f = [3.6\ 0.4\ 4\ 1.6]^T$, the firing rate for each transition is 1, and the sampling period $\Theta = 0.2$. One minimal firing count vector to reach $\boldsymbol{m}_f$ (in this case, the unique one) is $\boldsymbol{\sigma} = [0.4\ 0\ 4\ 0]^T$. By applying the ON/OFF+ controller, at each time step the firing flow of t_3 is forced to be 10 times the flow of t_1. For instance, at the first time step, $[0.04\ 0\ 0.4\ 0]^T$ is fired, reaching making $[7.56\ 0.04\ 0.4\ 1.96]^T$, etc. In this way, $\boldsymbol{m}_f$ is reached in 12 time steps. However, if t_3 fires before t_1, $\boldsymbol{m}_f$ can be reached in only 10 time steps (that is actually the minimum-time). In particular, at each of the first 9 time steps only t_3 fires, i.e., $[0\ 0\ 0.4\ 0]^T$ is fired; at the last time step, t_1 and t_3 fire, i.e., $[0.4\ 0\ 0.4\ 0]^T$ is fired.*

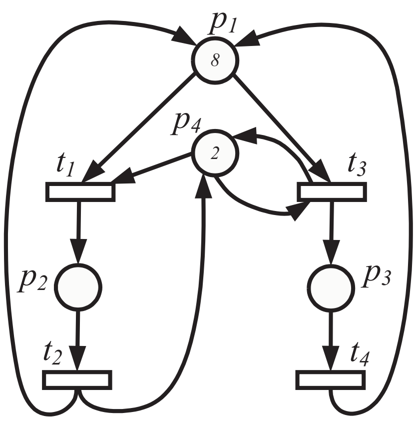


Figure 3: An EQ PN system with $\boldsymbol{m}_0 = [8\ 0\ 0\ 2]^T$.

Remark 4.6. *The results of Proposition 4.4 can be naturally extended to continuous-time CPN by making the sampling period Θ tend to 0.*

4.2 Balanced ON/OFF controller (B-ON/OFF)

We can apply the ON/OFF+ controller to any TCPN system and ensure the convergence to a final state $\boldsymbol{m}_f > 0$. Extremely fast to compute, nevertheless a possible drawback of this method is the following: since a set of transitions in coupled conflict relation are forced to fire proportionally, the required number of time steps for firing $\boldsymbol{\sigma}$ is determined by the "slowest" ones. Therefore, in extreme cases, when some of these transitions have very small (uncontrolled) flows, the whole system may be slowed down.

Example 4.7. *Let us consider the simple (sub-)system in Fig.4, assuming that t_1, t_2 have the same firing rate equal to 1. Moreover, they are forced by a given $\boldsymbol{\sigma}$ to fire in the same amounts. It is obvious that the flow of t_2 is 100 times the flow of t_1, but if t_1 and t_2 should fire proportionally according to $\boldsymbol{\sigma}$, then t_2 is slowed down.*

To overcome extremely bad cases, we can fire first the "faster" transitions and block the "slower" ones for some time periods, expecting that the flows (speeds) of the "slower" transitions are increased, i.e., we will try to *balance* the "faster" and "slower" transitions. After that, we simply apply the pure ON/OFF+ controller until the final state is reached.

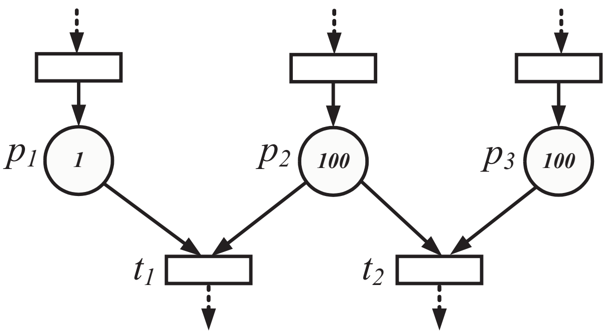


Figure 4: Fast transitions may be slowed down

We will first show how to classify the "slower" and "faster" transitions, and then present this balancing strategy.

Assume that the system is at marking $\boldsymbol{m}$ with $\boldsymbol{w}$ its controlled flow, and let $\boldsymbol{\sigma}$ be the firing count vector that should be fired to reach $\boldsymbol{m}_f$. Then $s_j = \lceil \frac{\boldsymbol{\sigma}[t_j]}{\boldsymbol{w}[t_j]\cdot\Theta} \rceil$ can be viewed as an estimation of the number of steps that transition t_j needs to fire, assuming that t_j fires with a constant speed equal to $\boldsymbol{w}[t_j]$. For two transitions t_a and t_b, if $s_a > s_b$, it is said that t_a is "slower" than t_b.

The estimation of the number of steps for t_j at $\boldsymbol{m}_0$ is defined by:

$$s_j^0 = \left\lceil \frac{\boldsymbol{\sigma}[t_j]}{\lambda_j \cdot enab(t_j, \boldsymbol{m}_0) \cdot \Theta} \right\rceil \tag{10}$$

If $enab(t_j, \boldsymbol{m}_0) = 0$ then $s_j^0 = \infty$.

Let us consider again the system in Ex.4.7 and let $\boldsymbol{\sigma}[t_1] = \boldsymbol{\sigma}[t_2] = 10$, $\Theta = 0.01$. The initial estimation of the number of time steps is: $s_1^0 = \frac{10}{1\cdot 0.01} = 1000$, $s_2^0 = \frac{10}{100\cdot 0.01} = 10$. So transition t_1 is "slower" than transition t_2.

Based on this initial estimation, we will partition any given set of transitions T_c that are in coupled conflict relation into two subsets, the "faster" ones T_1 and the "slower" ones T_2, such that:

$$\begin{cases} T_1 \cap T_2 = \emptyset, T_1 \cup T_2 = T_c \\ \forall t_a \in T_1, t_b \in T_2, s_b^0/s_a^0 > d \\ \forall t_{a1}, t_{a2} \in T_1, s_{a1}^0/s_{a2}^0 \leq d \end{cases} \tag{11}$$

where $d \geq 1$ is a design parameter used to classify "slower" and "faster" transitions.

From (11), the estimations of the number of time steps of the transitions in T_2 are at least d times as large as the ones of transitions in T_1. If we fire the transitions in T_1 and T_2 proportionally, transitions in T_1 may be slowed down by the ones in T_2.

Notice that, if the value of d is too large, all the transitions are put into T_1, then it is equivalent to applying the ON/OFF+ controller directly; if d is too small, most of the transitions are put into T_2 and initially blocked. In the system shown in Ex.4.7, we can choose, for example, $d = 10$. Then the conflicting transition set $T_c = \{t_1, t_2\}$ is partitioned to $T_1 = \{t_2\}$ and $T_2 = \{t_1\}$.

Now let us consider that the system is at time step k with marking $\boldsymbol{m}_k$, and the firing count vector $\boldsymbol{\sigma'}$ has been fired, i.e., $\boldsymbol{m}_k = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma'}$. The remaining firing count vector that should be fired is $\boldsymbol{\sigma}_k = \boldsymbol{\sigma} - \boldsymbol{\sigma'} \geq 0$. The estimation of the number of steps for transition $t_j \in T_c$ at $\boldsymbol{m}_k$ is defined by:

$$s_j^k = \begin{cases} \lceil \frac{\boldsymbol{\sigma}_k[t_j]}{\boldsymbol{w}_k[t_j]\cdot\Theta} \rceil, & \text{if } t_j \in T_1 \\ \lceil \frac{\boldsymbol{\sigma}_k[t_j]}{\lambda_j \cdot enab(t_j, \boldsymbol{m}_k)\cdot\Theta} \rceil, & \text{if } t_j \in T_2 \end{cases}$$

where $\boldsymbol{w}_k[t_j]$ is the flow of transition t_j when the ON/OFF+ strategy is applied. Because the transitions in T_1 fire proportionally, for any $t_j \in T_1$, the same estimation s_j^k is obtained, denoted by h^k. For any $t_b \in T_2$, let $D_b^k = s_b^k/h^k$, which reflects the difference of the estimations between t_b and the faster transitions.

Let T_p be the set of persistent transitions, and T_c^i, $i = 1, 2, 3, ..., l$ be the sets of transitions in coupled conflict. Algorithm 3 synthesizes the control method: for transitions in T_p, the ON/OFF strategy is always applied; for any $T_c^i = T_1^i \cup T_2^i$, those "faster" transitions in T_1^i fire proportionally using the ON/OFF+ strategy; while every "slower" transition t_b in T_2^i is blocked until the following condition (C1) or (C2) is satisfied; then we move t_b to T_1^i and start to fire it using the ON/OFF+ strategy.

(C1) $D_b^k \leq d$

(C2) $D_b^k \geq D_b^{k-1}$

By blocking t_b while firing other transitions, more tokens may arrive to the input places of t_b, consequently increasing its flow. Thus, t_b may become more balanced with those "faster" transitions, i.e., D_b^k decreases. If D_b^k keeps decreasing, for sure in finite time, we will have condition (C1) satisfied, implying that t_b is already balanced with the "faster" transitions. If at one moment, D_b^k cannot decrease any more, then condition (C2) is satisfied, i.e., transition t_b cannot become more balanced with the "faster" ones. Therefore, one of these conditions will be satisfied in finite time. After that, there is no need to block t_b and we should start to fire it.

Now we prove the convergence of this B-ON/OFF controller to the desired final state.

Proposition 4.8. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a TCPN system with $\boldsymbol{m}_0 > 0$. Let $\boldsymbol{m}_f > 0$ be a reachable final marking, such that $\boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$. By applying the B-ON/OFF controller, the system state converges to $\boldsymbol{m}_f$ in finite time.*

Proof: For all the "slower" transitions, condition (C1) or (C2) will be satisfied in a finite number of steps, then the pure ON/OFF+ strategy is applied. Therefore, we only need to prove that when the pure ON/OFF+ controller starts to be applied, the system is in a state $\boldsymbol{m} > 0$ and $\boldsymbol{m}_f$ is reachable from $\boldsymbol{m}$.

Since the accumulative firing counts of transitions are upper bounded by $\boldsymbol{\sigma}$, then we have $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}'$, $0 \leq \boldsymbol{\sigma}' \leq \boldsymbol{\sigma}$. Because $\boldsymbol{m}_0 > 0$, in a finite time $\boldsymbol{m} > 0$. Since $\boldsymbol{\sigma} - \boldsymbol{\sigma}' \geq 0$ and $\boldsymbol{m}_f = \boldsymbol{m} + \boldsymbol{C} \cdot (\boldsymbol{\sigma} - \boldsymbol{\sigma}') > 0$, $\boldsymbol{m}_f$ is reachable from $\boldsymbol{m}$ ([7]). ■

Remark 4.9. *The B-ON/OFF controller is more computationally expensive than the ON/OFF+ controller, because an estimation of the number of time steps has to be computed at each step. However, the B-ON/OFF strategy may significantly decrease the time of reaching the final state if the flows of conflicting transitions are of different orders of magnitude. The choice of design parameter* d *also influences the performance. In particular, if* d *is too large, the controller is not "sensitive" to the difference between "faster" and "slower" transitions, thus it is similar to applying the ON/OFF+ strategy. Nevertheless, this does not mean that* d *should always be as small as possible, because if the flows of conflicting transitions are similar, it may make sense to apply the ON/OFF+ strategy.*

4.3 MPC-ON/OFF controller

Both the ON/OFF+ and B-ON/OFF controllers solve the conflict based on the flows and the required firing counts in a current time step, without a "careful looking at the future". In this section, we combine the ON/OFF strategy with Model Predictive Control (MPC), obtaining the MPC-ON/OFF controller.

MPC has been widely applied in the industry for controlling complex dynamic systems. By solving a discrete-time optimal control problem over a given time horizon, an optimal open-loop control input sequence is obtained and the first one is applied. Then at the next time step, a new optimal control

Algorithm 3 B-ON/OFF Controller

Input: $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$, $\boldsymbol{m}_f$, $\boldsymbol{\sigma}$, Θ, d , T_p, T_c^i, $i = 1, 2, 3..., l$

Output: $\boldsymbol{w}_0, \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots$

Partition every T_c^i into T_1^i and T_2^i, $i = 1, 2, ..., l$

$k \leftarrow 0$

while $\sum_{i=0}^{k-1} \boldsymbol{w}_i \cdot \Theta \neq \boldsymbol{\sigma}$ **do**

  Obtain $\boldsymbol{w}_k[t_j]$ for any $t_j \in T_p$: applying the ON/OFF strategy

  for $i = 1$ **to** l **do**

    For any $t_j \in T_2^i$: $\boldsymbol{w}_k[t_j] \leftarrow 0$

    Obtain $\boldsymbol{w}_k[t_j]$ for any $t_j \in T_1^i$: applying the ON/OFF+ strategy

  end for

  Apply $\boldsymbol{w}_k : \boldsymbol{m}_{k+1} \leftarrow \boldsymbol{m}_k + \boldsymbol{C} \cdot \boldsymbol{w}_k \cdot \Theta$

  $\boldsymbol{\sigma}_{k+1} \leftarrow \boldsymbol{\sigma} - \sum_{i=0}^{k} \boldsymbol{w}_i \cdot \Theta$

  for $i = 1$ **to** l **do**

    if $T_2^i \neq \emptyset$ **then**

      Compute $\boldsymbol{w}_{k+1}[t_a], t_a \in T_1^i$

      $h^{k+1} \leftarrow \boldsymbol{\sigma}_{k+1}[t_a]/(\boldsymbol{w}_{k+1}[t_a] \cdot \Theta)$

      for each $t_b \in T_2^i$ **do**

        $s_b^{k+1} \leftarrow \boldsymbol{\sigma}_{k+1}[t_b]/(\lambda_b \cdot enab(t_b, \boldsymbol{m}_{k+1}) \cdot \Theta)$

        $D_b^{k+1} \leftarrow s_b^{k+1}/h^{k+1}$

        if $D_b^{k+1} \leq d$ **or** $D_b^{k+1} \geq D_b^k$ **then**

          $T_1^i \leftarrow T_1^i \cup \{t_b\}$

          $T_2^i \leftarrow T_2^i \setminus \{t_b\}$

        end if

      end for

    end if

  end for

  $k \leftarrow k + 1$

end while

return $\boldsymbol{w}_0, \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots$

problem is solved. In [13], the MPC scheme is applied to the control of TCPN, by solving the following optimization problem at each time step:

$$\begin{aligned}
\min \quad & J(\boldsymbol{m}_k, N) \\
s.t. : \quad & \boldsymbol{m}_{k+j+1} = \boldsymbol{m}_{k+j} + \Theta \cdot C \cdot \boldsymbol{w}_{k+j}, j = 0, ..., N-1 && (12a) \\
& \boldsymbol{G} \cdot \begin{bmatrix} \boldsymbol{w}_{k+j} \\ \boldsymbol{m}_{k+j} \end{bmatrix} \leq 0, j = 0, ..., N-1 && (12b) \\
& \boldsymbol{w}_{k+j} \geq 0, j = 0, ..., N-1 && (12c)
\end{aligned}$$

where $J(\boldsymbol{m}_k, N)$ may be a quadratic objective (cost) function in the form of (13), while $\boldsymbol{G}$ is a particular matrix deduced from the net structure and (12b) gives the (upper bound) constraint on firing flows to guarantee the non-negativeness of markings.

$$
\begin{aligned}
J(\boldsymbol{m}_k, N) = & (\boldsymbol{m}_{k+N} - \boldsymbol{m}_f)' \cdot \boldsymbol{Z} \cdot (\boldsymbol{m}_{k+N} - \boldsymbol{m}_f) \\
& + \sum_{j=0}^{N-1} [(\boldsymbol{m}_{k+j} - \boldsymbol{m}_f)' \cdot \boldsymbol{Q} \cdot (\boldsymbol{m}_{k+j} - \boldsymbol{m}_f) \\
& + (\boldsymbol{w}_{k+j} - \boldsymbol{w}_f)' \cdot \boldsymbol{R} \cdot (\boldsymbol{w}_{k+j} - \boldsymbol{w}_{\boldsymbol{f}})]
\end{aligned} \tag{13}
$$

MPC is usually used for optimizing trajectories subject to certain constraints. In our problem, the aim is to reach $\boldsymbol{m}_f$ as soon as possible, i.e., minimizing the time. Although it is difficult to obtain a minimum-time control by using a MPC approach, we will consider this method for transitions in conflicts while for the others we will keep the ON/OFF strategy. It will be shown that we may obtain smaller number of time steps than those of the ON/OFF+ or B-ON/OFF controller, usually with large time horizon and with higher computational complexity.

Let us denote by T_p the set of persistent transitions and T_c the set of transitions in any coupled conflict relation, $T_p \cap T_c = \emptyset, T_p \cup T_c = T$. The MPC-ON/OFF controller is synthesized in Algorithm 4.

Algorithm 4 MPC-ON/OFF controller

Input: $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$, $\boldsymbol{m}_f$, $\boldsymbol{\sigma}$, Θ, $\boldsymbol{Z}$, $\boldsymbol{Q}$, $\boldsymbol{R}$, N ϵ, ζ
Output: $\boldsymbol{w}_0$, $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, ...

```
k ← 0
σ_k ← σ
while m_k ≠ m_f do
    Solve problem (14)
    Apply w_k : m_{k+1} ← m_k + C · w_k · Θ
    σ_{k+1} ← σ_k − w_k · Θ
    k ← k + 1
end while
return w_0, w_1, w_2, ...
```

The problem that should be solved at each time step k is:

$$
\begin{aligned}
\min \quad & J(\boldsymbol{m}_k, N) \\
s.t.: \quad & \boldsymbol{m}_{k+j+1} = \boldsymbol{m}_{k+j} + C \cdot \boldsymbol{w}_{k+j} \cdot \Theta, j = 0, ..., N-1 && \text{(14a)} \\
& \boldsymbol{G} \cdot \begin{bmatrix} \boldsymbol{w}_{k+j} \\ \boldsymbol{m}_{k+j} \end{bmatrix} \leq 0, j = 0, ..., N-1 && \text{(14b)} \\
& \boldsymbol{w}_{k+j} \geq 0, j = 0, ..., N-1 && \text{(14c)} \\
& \sum_{j=0}^{N-1} \boldsymbol{w}_{k+j} \cdot \Theta \leq \boldsymbol{\sigma}_k && \text{(14d)} \\
& \boldsymbol{m}_{k+1} \geq \boldsymbol{1} \cdot \epsilon && \text{(14e)} \\
& \boldsymbol{1}^T \cdot \boldsymbol{w}_k \geq \zeta && \text{(14f)}
\end{aligned}
$$

where ϵ and ζ are sufficient small positive numbers and $\boldsymbol{\sigma}_k$ is the remaining firing count vector that should be fired. Constraint (14e) ensures that the system only evolves inside an interior region of the reachability space; in order to include $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ in that region, it should hold $\boldsymbol{m}_f \geq \boldsymbol{1} \cdot \epsilon$ and $\boldsymbol{m}_0 \geq \boldsymbol{1} \cdot \epsilon$. Constraint (14f) forces a non-zero flow in the first predictive step. For our specific problem, we use the following assumptions:

(1) $\boldsymbol{m}_0, \boldsymbol{m}_f > 0$.

(2) $\boldsymbol{Z}, \boldsymbol{Q} \in \mathbb{R}^{|P|} \geq 0$ are positive definite matrices.

(3) $\boldsymbol{R} \in \mathbb{R}^{|T|}_{\geq 0}$ is a diagonal matrix, such that if $t_j \in T_p$, $\boldsymbol{R}[j,j] > 0$, otherwise $\boldsymbol{R}[j,j] = 0$.

We define the cost function as:

$$\begin{aligned} J(\boldsymbol{m}_k, N) = & (\boldsymbol{m}_{k+N} - \boldsymbol{m}_f)' \cdot \boldsymbol{Z} \cdot (\boldsymbol{m}_{k+N} - \boldsymbol{m}_f) \\ & + \sum_{j=0}^{N-1} [(\boldsymbol{m}_{k+j} - \boldsymbol{m}_f)' \cdot \boldsymbol{Q} \cdot (\boldsymbol{m}_{k+j} - \boldsymbol{m}_f)] \\ & - \boldsymbol{w}'_k \cdot \boldsymbol{R} \cdot \boldsymbol{w}_k \end{aligned} \tag{15}$$

By means of the item $-\boldsymbol{w}'_k \cdot \boldsymbol{R} \cdot \boldsymbol{w}_k$ in the cost function and choosing large values for $\boldsymbol{R}[j,j], t_j \in T_p$, we try to fire the persistent transitions as fast as possible, similarly to applying the ON/OFF strategy. Now, we will prove that the asymptotic stability holds.

Proposition 4.10. *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a TCPN system with $\boldsymbol{m}_0 > 0$. Let $\boldsymbol{m}_f > 0$ be a reachable final marking, such that $\boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$. Assume that the system is controlled by using the MPC-ON/OFF controller shown in Algorithm 4, and the weighting matrices $\boldsymbol{Z}, \boldsymbol{Q}, \boldsymbol{R}$ satisfy the assumptions. Then the closed-loop system is asymptotically stable.*

Proof: We will define a quadratic Lyapunov function and prove that it is strictly decreasing.

Let $V(\boldsymbol{m}_k) = \boldsymbol{1}^T \cdot (\boldsymbol{\sigma} - \sum_{i=0}^{k} \boldsymbol{w}_i \cdot \Theta)$, where $\boldsymbol{w}_k$ is the controlled flow at time step k and Θ is the sampling period. According to constraint (14d), the accumulative firing count is upper bounded by $\boldsymbol{\sigma}$. Therefore, $V(\boldsymbol{m}_k) \geq \boldsymbol{0}$ and $V(\boldsymbol{m}_k) \neq 0$ until $\boldsymbol{\sigma} = \sum_{i=0}^{k} \boldsymbol{w}_i \cdot \Theta$, i.e., until $\boldsymbol{m}_k = \boldsymbol{m}_f$. Now we need to prove that $V(\boldsymbol{m}_k)$ is strictly decreasing, and it is equivalent to prove that $\boldsymbol{w}_k \neq 0$ until $\boldsymbol{\sigma}$ is reached. Considering the last constraint (14f), we only need to prove that problem (14) is feasible until $\boldsymbol{m}_f$ is reached.

Assume that the system is at time step k with marking $\boldsymbol{m}_k \neq \boldsymbol{m}_f$, according to constraint (14e), we have $\boldsymbol{m}_k > 0$. Let us denote by $\boldsymbol{\sigma'}$ the firing count vector that has been fired. It is clear that $\boldsymbol{\sigma'} \leq \boldsymbol{\sigma}$, therefore, $\boldsymbol{\sigma} - \boldsymbol{\sigma'} \geq 0$ and:

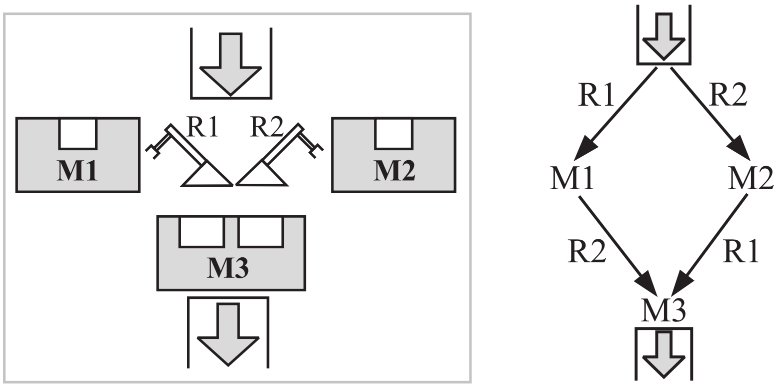
$$\boldsymbol{m}_f = \boldsymbol{m}_k + \boldsymbol{C} \cdot (\boldsymbol{\sigma} - \boldsymbol{\sigma'}) > 0 \tag{16}$$

so $\boldsymbol{m}_f$ is reachable from $\boldsymbol{m}_k$ [7]. Since the reachability space of CPN is a convex set [20], the states on the straight line from $\boldsymbol{m}_k$ to $\boldsymbol{m}_f$ are reachable. Because $\boldsymbol{m}_k \geq \boldsymbol{1} \cdot \epsilon > 0$ (constraint (14e)), all the transitions are enabled at $\boldsymbol{m}_k$. There exists $\alpha > 0$, $\boldsymbol{w}_k = \alpha \cdot (\boldsymbol{\sigma} - \boldsymbol{\sigma'}) \neq 0$, such that in the next predictive step the system could reach a state $\boldsymbol{m}_{k+1}$ that is on the straight path from $\boldsymbol{m}_k$ to $\boldsymbol{m}_f$. Since $\boldsymbol{m}_k, \boldsymbol{m}_f \geq \boldsymbol{1} \cdot \epsilon$, we have $\boldsymbol{m}_{k+1} \geq \boldsymbol{1} \cdot \epsilon$. Then let $\boldsymbol{w}_k = \alpha \cdot (\boldsymbol{\sigma} - \boldsymbol{\sigma'}) \neq 0$, if ζ is small enough (e.g., the epsilon machine) the problem (14) is feasible until $\boldsymbol{m}_f$ is reached. ∎

5 An application example

In this section, we illustrate the proposed methods by means of a flexible manufacturing system. The simulations are performed by using MATLAB on a PC with Intel(R) Core(TM)2 Quad CPU Q9400 @ 2.66GHz, 3.24GB of RAM.

Let us consider the manufacturing cell shown in Fig. 5. It consists of three machines M1, M2, M2 and two robots R1, R2. Two semi-products A and B are processed by M1 and M2, respectively, then they are assembled in M3 to get the final product. Robot R1 moves the raw materials from the input buffer to M1 or moves the semi-product B from M2 to M3; robot R2 moves the materials from the input



(a) (b)

Figure 5: (a) Logical layout of a manufacturing cell (b) Its production process

buffer to M2 or moves the semi-product A from M1 to to M3. The logical layout and production process are demonstrated in Fig. 5(a) and Fig. 5(b).

The PN model of the described manufacturing cell is presented in Fig. 6. When machine M1 is available (p_{14} is marked) and robot R1 is idle (p_{18} is marked), a part of raw material for semi-product A can be loaded to machine M1, changing to loading state (p_1). When the loading process finishes (t_2 fires), M1 changes to working state (p_2) and robot R1 is freed (p_{18} is marked). Transition t_3 models the working process of machine M1, and when it finishes, the semi-product A is stored in p_3. If the slots for semi-product A in machine M3 (p_{12}) is available and robot R2 is idle, the semi-product A can be loaded from machine M1 to machine M3 then waits in p_9. The behavior of M2 for processing semi-product B is modeled in a similar way, but the robots are used in a different order. Finally, if both semi-products A (in p_9) and B (in p_{10}) are available, they are assembled (*rendez-vous*) to the final product. The meaning of the places and transitions of the PN model is in Table 1.

We assume that each robot can only handle one piece of raw materials or semi-products and machine M1, M2 can accept maximally two pieces at the same time; machine M3 has 4 free slots for each type of semi-products; and it is assumed that we initially have 5 pieces of raw materials. According to this setting, the initial state $\boldsymbol{m}_0$ of the system is described in Fig. 6.

Considering the system as timed, we assume that every transition t_j has an average delay time, denoted by $\boldsymbol{\delta}[t_j]$. In particular, the loading time of raw materials to machine M1 and M2 are all equal to 0.1 time units, i.e., $\boldsymbol{\delta}[t_1] = \boldsymbol{\delta}[t_2] = \boldsymbol{\delta}[t_6] = \boldsymbol{\delta}[t_7] = 0.1$; the loading operations of semi-products from machines M1 to M3 and M2 to M3 take 0.4 time units, i.e., $\boldsymbol{\delta}[t_4] = \boldsymbol{\delta}[t_5] = \boldsymbol{\delta}[t_9] = \boldsymbol{\delta}[t_{10}] = 0.4$; the processing of machine M1 requires 0.5 time units, and for M2, it is 0.8 time units, i.e., $\boldsymbol{\delta}[t_3] = 0.5, \boldsymbol{\delta}[t_8] = 0.8$; it takes 0.4 time units to combine the two semi-products, and 2 time units to process them in machine M3, i.e., $\boldsymbol{\delta}[t_{11}] = 0.4, \boldsymbol{\delta}[t_{12}] = 2$. In the corresponding TCPN model under infinite sever semantics, time delays are approximated by their mean values ($\boldsymbol{\lambda}[t_j] = 1/\boldsymbol{\delta}[t_j]$, $t_j \in T$), obtaining a deterministic approximation of the discrete case [18].

Let us consider the target marking control problem of this system. In order to have a positive initial marking needed by the control methods, we assume that all the emptied places in Fig. 6 have initial marking equal to 0.1. For a manufacturing system, normally we want to maximize the throughput (flow) of the system. In particular, we can verify that this system has a unique minimal T-semiflow equal to $\mathbf{1}$; therefore, it is equivalent to maximize the flow of any transition. By solving a LPP proposed in [14] for optimal steady-state control problem, the maximal flow is $\psi = 0.775$. With $\psi = 0.775$, we can compute a final marking state $\boldsymbol{m}_f$ with the minimal Work In Process (WIP) cost, by solving the following LPP

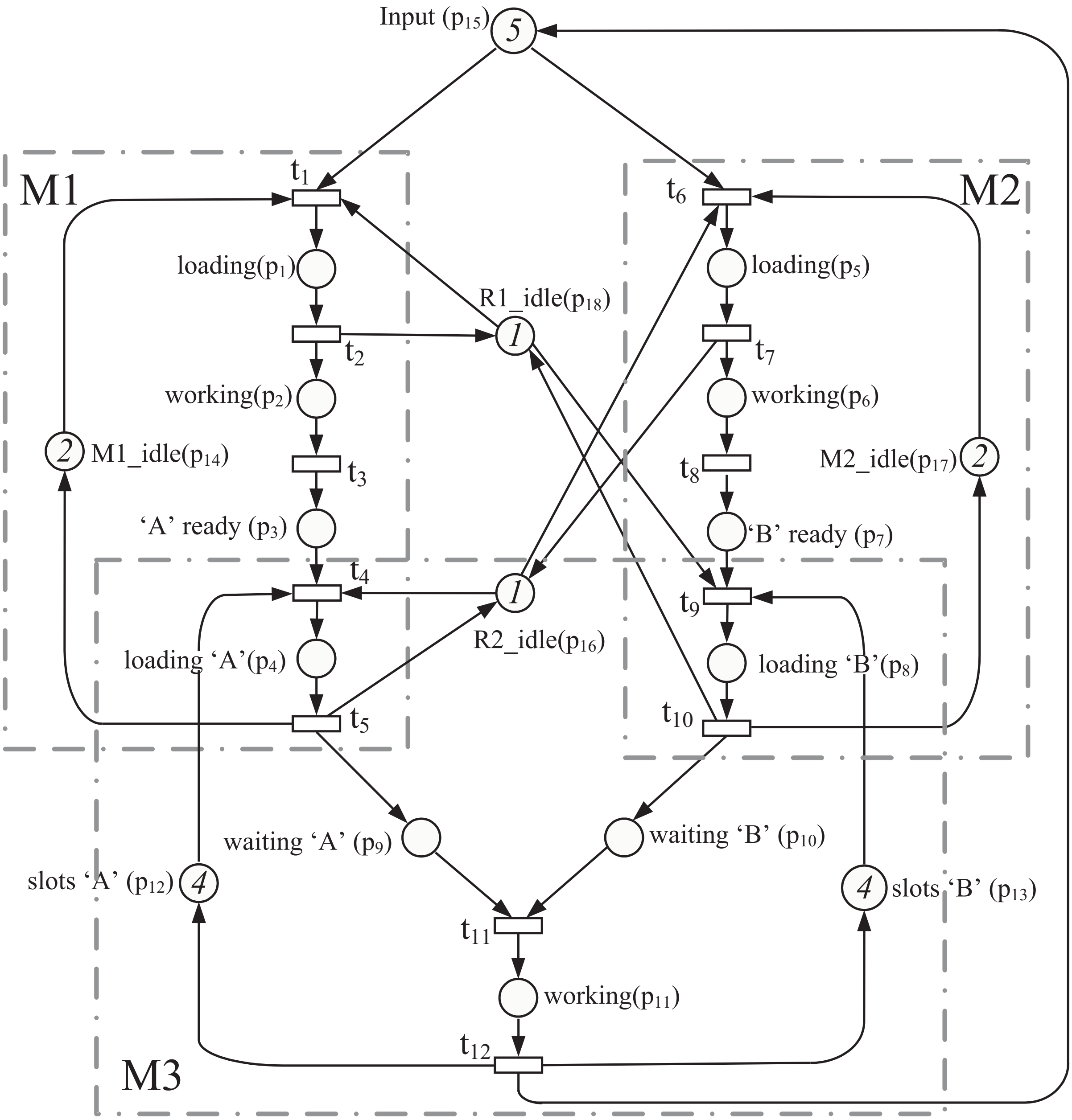


Figure 6: The PN model of the manufacturing cell

problem:

Table 1: The interpretation of the PN model in Fig. 6

Place	Interpretation	Transition	Interpretation
p_1	M1 is loading	t_1	R1 starts to load M1
p_2	M1 is working	t_2	R1 loading finishes
p_3	semi-product A is ready	t_3	M1 finishes processing
p_4	M3 is loading A	t_4	R2 starts to load A to M3
p_5	M2 is loading	t_5	R2 loading A finishes
p_6	M2 is working	t_6	R2 starts to load M2
p_7	semi-product B is ready	t_7	R2 loading finishes
p_8	M3 is loading B	t_8	M2 finishes processing
p_9	A is waiting for assembling	t_9	R1 starts to load B to M3
p_{10}	B is waiting for assembling	t_{10}	R1 loading B finishes
p_{11}	M3 is working	t_{11}	combine A and B
p_{12}	available slots for A	t_{12}	assembling finishes
p_{13}	available slots for B		
p_{14}	M1 is available		
p_{15}	input raw materials		
p_{16}	R2 is idle		
p_{17}	M2 is available		
p_{18}	R1 is idle		

$$
\begin{array}{ll}
\min & \boldsymbol{l} \cdot \boldsymbol{m}_f \\
\text{s.t.} & \boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma} \\
& \boldsymbol{C} \cdot \boldsymbol{w}_f = \boldsymbol{0} \\
& \boldsymbol{w}_f[t] = \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{m}_f[p_i]}{\boldsymbol{Pre}[p_i,t]} - \boldsymbol{v}[p_i,t], \ \forall t \in T, p_i \in {}^{\bullet}t \\
& \boldsymbol{v}[p_i,t] \geq 0 \\
& \boldsymbol{w}_f[t_j] = \psi, \ \forall t_j \in T \\
& \boldsymbol{w}_f, \boldsymbol{\sigma}, \boldsymbol{m}_f \geq \boldsymbol{0}
\end{array}
\quad (17)
$$

where $\boldsymbol{v}[p_i,t]$ are slack variables, $\boldsymbol{w}_f$ is the controlled flow in final state $\boldsymbol{m}_f$ and $\boldsymbol{l}$ is the cost vector due to immobilization to maintain the production flow, e.g., due to the levels in stores. In this example, let us assume that we try to minimize the storage, i.e., the number of tokens, in the buffer places, i.e., $\boldsymbol{l}[p_3] = \boldsymbol{l}[p_7] = \boldsymbol{l}[p_9] = \boldsymbol{l}[p_{10}] = 1$ and for other places p_i, let $\boldsymbol{l}[p_i] = 0$. By solving LPP (17), we obtain an optimal final state $\boldsymbol{m}_f = [0.0775\ 0.3875\ 0.31\ 0.31\ 0.0775\ 0.62\ 0.31\ 0.31\ 0.31\ 0.31\ 1.55\ 2.13\ 2.13\ 1.315\ 0.0775\ 0.8125\ 1.083\ 0.8125]^T$, such that the maximal flow and minimal WIP cost are achieved.

The existence of several control methods for the target marking control problem of TCPN makes difficult the selection of the most appropriate technique for a given system. In order to have a good "guess", several properties may be taken into account for selecting a proper control method, e.g., feasibility, closed-loop stability, robustness, computational complexity (for the synthesis and during the execution). Table 2 shows qualitative characteristics of the already mentioned control methods. Apart from the methods proposed in this work, the approaching minimal-time (app. min-time) controller proposed in [1] and the MPC controller proposed in [13] are also included in the comparison.

The ON/OFF controller is particularly suitable for the minimum-time control of CF nets, while all the other methods can be applied to general net systems. Except the MPC controller, the others are heuristic approaches with the objective of reaching the desired final state in minimum-time; the MPC controller is usually used to optimize the state trajectory by minimizing a quadratic or linear cost function. For the ON/OFF, ON/OFF+ and B-ON/OFF controller, in each step only a LPP needs to be solved, therefore

Table 2: Qualitative characteristics of control methods (assuming $\boldsymbol{m}_0 > 0$, $\boldsymbol{m}_f > 0$). The following abbreviations are used: min. (minimize), func. (function), suff. (sufficient conditions), quad. (quadratic) and poly. (polynomial)

Methods	**Subclass**	**Computational issues**	**Optimizing index**	**Stability**
MPC [13]	All	Poly. on $\|T\|, N$	Quad. or linear func.	under suff.
App. min-time [1]	All	Nonlinear	Heuristic Min. Time	yes
ON/OFF [31]	CF	Poly. on $\|T\|$	Min. time	yes
ON-OFF+	All	Poly. on $\|T\|$	Heuristic Min. Time	yes
B-ON/OFF	All	Poly. on $\|T\|$	Heuristic Min. Time	yes
MPC-ON/OFF	All	Poly. on $\|T\|$, N	Heuristic Min. Time	yes

those methods have very low computational complexity. Nevertheless, for the MPC and MPC-ON/OFF controller, the number of variables also depends on the time horizon N, being computationally expensive if N is large. The approaching minimum-time controller also has high computational complexity, since non-linear problems have to be solved when intermediate states are added to the trajectory for decreasing the duration of the evolution.

The system is not CF, so the standard ON/OFF controller may not be applicable. Table 3 compares the number of time steps required for reaching $\boldsymbol{m}_f$ by using the other described control methods, and the corresponding CPU time for computing the control laws (in case that some free parameters exist, they are chosen for obtaining relatively smaller number of time steps).

Table 3: Comparison of different control methods for reaching $\boldsymbol{m}_f$ of the TCPN model in Fig. 6: time steps and CPU computing time

Method	Time steps	CPU time (ms)	Parameters
MPC(N=1)	230	1713	$\boldsymbol{Q} = 1000 \cdot \boldsymbol{I}^{\|P\|}$, $\boldsymbol{R} = 0.1 \cdot \boldsymbol{I}^{\|T\|}$, $\boldsymbol{Z} = \boldsymbol{P}$, the solution of the unconstrained LQR [22]
MPC(N=5)	223	21936	same as the above
App. min-time	335	37495	threshold = 0.001
ON/OFF+	457	225	none
B-ON/OFF	209	324	$d = 3$
MPC-ON/OFF(N=1)	220	2615	$\boldsymbol{Z} = 1000 \cdot \boldsymbol{I}^{\|P\|}$, $\boldsymbol{Q} = \boldsymbol{I}^{\|P\|}$, $R(j,j) = 1000, \forall t_j \in T_p$, $\epsilon = \zeta = 10^{-5}$
MPC-ON/OFF(N=5)	218	72205	same as the above

$\star$ $\boldsymbol{m}_0 = [0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 4\ 4\ 2\ 5\ 1\ 2\ 1]^T$, $\boldsymbol{m}_f = [0.0775\ 0.3875\ 0.31\ 0.31\ 0.0775\ 0.62\ 0.31\ 0.31\ 0.31\ 0.31\ 1.55\ 2.13\ 2.13\ 1.315\ 0.0775\ 0.8125\ 1.083\ 0.8125]^T$, $\boldsymbol{\lambda} = [10\ 10\ 2\ 2.5\ 2.5\ 10\ 10\ 1.25\ 2.5\ 2.5\ 2.5\ 0.5]^T$, $\boldsymbol{\sigma} = [2.345\ 2.368\ 2.08\ 1.87\ 1.66\ 2.578\ 2.6\ 2.08\ 1.87\ 1.66\ 1.45\ 0]^T$, $\Theta = 0.01$

Table 3 shows that the B-ON/OFF gives the smallest number of time steps for reaching the final state and the computational complexity is also very low. Consider the ON/OFF+ controller: the flows of t_4 and t_9 are much smaller than the ones of t_1 and t_6; those four transitions are in a coupled conflict, therefore if we fire them proportionally by using the ON/OFF+ strategy the result is not good, costing 457 time steps, more than the double of the B-ON/OFF controller, to reach $\boldsymbol{m}_f$. By applying the MPC-ON/OFF controller, even with a very small time horizon ($N = 1$), a reasonably small number of time steps (220) is obtained and, in this case, the CPU time is not very high either; meanwhile, the number of time steps can be decreased (only 0.9%) by using a larger time horizon ($N = 5$), but the CPU time increases significantly (about 28 times). We can also apply the MPC controller, but the number of time steps here is larger than of the B-ON/OFF controller and MPC-ON/OFF controller; similarly, if we use a larger time horizon, the computational cost increases significantly but the number of times steps for reaching $\boldsymbol{m}_f$ only decreases slightly. Therefore, it seems that the interest of using a large N is not high, and we "guess" the importance of using a small time horizon for the MPC approaches. On the other hand, let us compare the MPC controller with the MPC-ON/OFF controller, for example, with $N = 1$: the MPC-ON/OFF controller can decrease the number of times steps of the MPC controller for more than 4%; meanwhile, with only about 1.5 times its computational cost. The approaching minimum-time (app. min-time) controller does not work very well in this case: 1) 335 steps are needed. One reason may be that in this approach, between each pair of adjacent states of the trajectory the firing speed is constant and determined by the one with smaller flow; therefore, if one of the states has very small flow (in this case, the initial one), the time spent for reaching $\boldsymbol{m}_f$ could be large; 2) The required CPU time of the approaching minimum-time controller is also quite high, due to the non-linear problems that should be solved. Nevertheless, notice that we just show the results of different methods for a particular example to provide a educated "guess", but nothing concluding.

Last but not least, it may be interesting to point out that in "slow" practical systems like logistics, chemical or manufacturing systems, the operational time may be much larger than the computational time for the control laws (considering the very low computational complexity of ON/OFF based methods; including the MPC-ON/OFF controller with small time horizon); therefore, at each operation instant we could compute the control laws by using several of the methods, and then choose the best one to apply.

6 Conclusions

In this work, we present three heuristic methods, based on the ON/OFF strategy, for the minimum-time control of TCPN. It is proved that all of them can drive a general TCPN system to a desired final state in finite time. An initial comparison is carried out among the described control methods, including the proposed controllers and two others from the literature. Finally, the control laws are applied to a manufacturing cell. The advantage of these ON/OFF based controllers is their low computational complexity, while reasonable numbers of time steps for reaching the final states can be obtained.

The standard ON/OFF controller is the most suitable choice for a CF net system, ensuring low computational complexity and minimum-time. For non-CF nets, some characteristics of the system may be helpful in choosing an appropriate method. In particular, if the flows of conflicting transitions are very different, the B-ON/OFF controller may obtain a smaller number of time steps than the ON/OFF+ controller; the approaching minimum-time controller may not be a good choice if there are some places with very small markings or many borders of regions should be crossed from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$, in those cases the ON/OFF based methods usually can achieve better results.

As a future work, we plan to enhance the comparison of the different existing controllers and for this aim, certain benchmarks with systems of different net structures, markings, firing rates are desirable. On the other hand, the comparison should be performed not only with the continuous net systems, but also for the underlying discrete ones. Since all the transitions are assumed to be controllable in the present work, a clear extension is to consider partially controllable systems.

References

[1] H. Apaydin-Ozkan, J. Júlvez, C. Mahulea, and M. Silva. Approaching Minimum Time Control of Timed Continuous Petri nets. *Nonlinear Analysis: Hybrid Systems*, 5(2):136–148, May 2011.

[2] J.E. Bobrow, S. Dubowsky, and JS Gibson. Time-optimal control of robotic manipulators along specified paths. *The International Journal of Robotics Research*, 4(3):3–17, 1985.

[3] AE Bryson and Y.C. Ho. *Applied Optimal Control, Revised Printing*. Hemisphere, New York, 1975.

[4] R. David and H.Alla. *Discrete, Continuous and Hybrid Petri Nets*. Springer, Berlin, 2010. 2nd edition.

[5] S. Hennequin, D. Lefebvre, and A. El-Moudni. Fuzzy control of variable speed continuous Petri nets. In *38th IEEE Conference on Decision and Control*, volume 2, pages 1352–1356, Phoenix, USA, Dec. 1999.

[6] X. Jing, L. Recalde, and M. Silva. Tracking control of timed continuous Petri net systems under infinite servers semantics. In *IFAC World Congress*, 2008.

[7] J. Júlvez, L. Recalde, and M. Silva. On reachability in autonomous continuous Petri net systems. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *Applications and Theory of Petri Nets 2003*, volume 2679, pages 221–240, Eindhoven, The Netherlands, June 2003.

[8] R. Kara, M. Ahmane, J.J. Loiseau, and S. Djennoune. Constrained regulation of continuous Petri nets. *Nonlinear Analysis: Hybrid Systems*, 3(4):738–748, 2009.

[9] S. Keerthi and E. Gilbert. Computation of minimum-time feedback control laws for discrete-time systems with state-control constraints. *IEEE Transactions on Automatic Control*, 32(5):432–435, 1987.

[10] Dimitri Lefebvre, Catherine Delherm, Edouard Leclercq, and Fabrice Druaux. Some contributions with Petri nets for the modelling, analysis and control of HDS. *Nonlinear Analysis: Hybrid Systems*, 1:451–465, 2007.

[11] Z. W. Li and M. C. Zhou. Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(1):38–51, 2004.

[12] George F. List and Mecit Cetin. Modeling traffic signal control using Petri nets. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):177–187, 2004.

[13] C. Mahulea, A. Giua, L. Recalde, C. Seatzu, and M. Silva. Optimal model predictive control of Timed Continuous Petri nets. *IEEE Trans. on Automatic Control*, 53(7):1731 – 1735, August 2008.

[14] C. Mahulea, A. Ramirez, L. Recalde, and M.Silva. Steady state control reference and token conservation laws in continuous petri net systems. *IEEE Transactions on Automation Science and Engineering*, 5(2):307–320, April 2008.

[15] C. Mahulea, L. Recalde, and M. Silva. Basic Server Semantics and Performance Monotonicity of Continuous Petri Nets. *Discrete Event Dynamic Systems: Theory and Applications*, 19(2):189 – 212, June 2009.

[16] Dale B. McDonaldi. Structurally Time Optimal Control for Linear Systems with Bounded Control and Bias Disturbance. In *American Control Conference*, pages 1964 – 1969, New York, July 2007.

[17] J. Merseguer and J. Campos. Software performance modeling using UML and Petri nets. *Performance Tools and Applications to Networked Systems*, pages 265–289, 2004.

[18] L. Recalde and M. Silva. Petri nets fluidification revisited semantics and steady state. *APII JESA*, 35(4):435–449, 2001.

[19] L. Recalde, E. Teruel, and M. Silva. Autonomous Continuous P/T systems. In J. Kleijn S. Donatelli, editor, *Application and Theory of Petri Nets 1999*, volume 1639 of *Lecture Notes in Computer Science*, pages 107–126. Springer, 1999.

[20] Laura Recalde, Enrique Teruel, and Manuel Silva. Autonomous Continuous P/T Systems. In *Proceedings of the 20th International Conference on Application and Theory of Petri Nets*, pages 107–126, London, UK, UK, 1999. Springer-Verlag.

[21] R. Roberto, R. Antonio, M. Jose Alejandro, and R. Javier. Control Of Metabolic Systems Modeled with Timed Continuous Petri Nets. In *Proceedings of the Workshops of the 31st International Conference on Application and Theory of Petri Nets and Other Models of Concurrency (ICATPN 2010) and of the 10th International Conference on Application of Concurrency to System Design (ACSD 2010)*, pages 87–102, Braga, Portugal, June 2010.

[22] P.O.M. Scokaert and J.B. Rawlings. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, 43(8):1163–1169, 1998.

[23] K. Shin and N. McKay. Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, 30(6):531–541, 1985.

[24] M. Silva, J. Júlvez, C. Mahulea, and C. Renato Vázquez. On fluidization of discrete event models: observation and control of continuous Petri nets. *Discrete Event Dynamic Systems: Theory and Applications*, 21(4):427–497, 2011.

[25] M. Silva and L. Recalde. On fluidification of Petri net models: from discrete to hybrid and continuous models. *Annual Reviews in Control*, 28(2):253–266, 2004.

[26] L. M. Sonneborn and F. S. Van Vleck. The bang-bang principle for linear control systems. *Journal of the Society for Industrial & Applied Mathematics, Series A: Control*, 2(2):151–159, 1964.

[27] C.R. Vázquez and M. Silva. Performance control of Markovian Petri nets via fluid models: A stock-level control example. In *IEEE International Conference on Automation Science and Engineering*, pages 30–36, 2009.

[28] C.R. Vázquez and M. Silva. Piecewise-linear constrained control for timed continuous Petri nets. In *48th IEEE Conference on Decision and Control*, pages 5714 – 5720, Shanghai, Dec. 2009.

[29] CR. Vázquez, JH. van Schuppen, and M. Silva. A modular-coordinated control for continuous Petri nets. In *18th IFAC World Congress*, Milan, Italy, August 2011.

[30] L. Wang, C. Mahulea, J. Júlvez, and M. Silva. Control of continuous Petri nets using ON/OFF based method. In *WODES12: 11th Int. Workshop on Discrete Event Systems*, Guadalajara, Mexico, October 2012.

[31] LW. Wang, C. Mahulea, J. Júlvez, and M. Silva. Minimum-time Control for Structurally Persistent Continuous Petri Nets. In *49th IEEE Conference on Decision and Control*, pages 2771–2776, Atlanta, Georgia USA, December 2010.