Control of continuous Petri nets using ON/OFF based method

Liewei Wang * Cristian Mahulea * Jorge Jülvez * Manuel Silva *

* The authors are with the Aragón Institute of Engineering Research (I3A), University of Zaragoza, Maria de Luna 1, 50018 Zaragoza, Spain {lwang, cmahulea, julvez, silva@unizar.es}.

Abstract: Continuous Petri Nets (CPN) can be used to approximate classical discrete Petri nets which suffer from the state explosion problem. In this paper we focus on the control of timed CPN (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 structurally persistent nets, and it is proved to be minimum-time. In this work the ON/OFF controller is extended to general TCPN, but in this case, the minimum-time evolution is not guaranteed. Three extensions are proposed, all of them are based on the ON/OFF strategy. Some comparisons of those controllers are given in terms of their applications to an assembly system.

1. INTRODUCTION

Petri Nets (PN) is a well known paradigm used for modeling, analysis, and synthesis of discrete event systems (DES). With strong facility to depict sequences, concurrency, conflicts and synchronizations, it is widely applied in the industry for the analysis of manufacturing, traffic, or software systems, for example. 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) [1, 9] are fluid approximations of classical discrete PN obtained by removing the integrality constraints, which means that the firing count vector and consequently the marking are no longer restricted to be in the naturals but relaxed into the non-negative real numbers. An important advantage of this relaxation is that more efficient algorithms are available for their analysis. In [10], the control methods are first achieved in the fluidified continuous model, after that they are applied back to control its original discrete system.

One of the important objectives in the control of CPN is to drive the system from an initial state \( m_0 \) to a desired final state \( m_f \), which is similar to the set-point control problem in a general continuous-state system. By considering the CPN as a relaxation of discrete systems, the continuous state can be viewed as the approximation of the average state in its original discrete system. Several approaches can be found in the literature for handling this control problem, for example, in [2, 5, 7]. Many of these works are based on infinite server semantics. For a broad class of nets, it has been proved that this semantics provides a better approximation of discrete systems than finite sever semantics under some general conditions [6]. In the case of systems with uncontrollable transitions, the control problem may become much more complex [4, 11].

A minimum-time ON/OFF controller has been proposed for structurally persistent PN [12]. The essential problem of this standard ON/OFF controller is that when there is conflict, this “greedy” strategy of firing transitions may bring the system to a “blocked” situation (see Ex.2 for an example). In this work, the ON/OFF control scheme is further investigated and three heuristic extensions are presented, ensuring that the final state is reached in finite time, even if the minimum time is not guaranteed. By forcing the conflicting transitions firing proportionally, we obtain the ON/OFF-plus (ON/OFF+) controller. But the drawback of this method is obvious: the firing speeds of transitions in a conflict relation are decided by the slower ones, and the overall system may be highly slowed down. Therefore, the second extension, balanced ON/OFF (B-ON/OFF) controller is proposed, trying to balance the fast and slow transitions before applying the pure ON/OFF+ controller. The third method is a combination of model predictive control (MPC) and ON/OFF strategy: solving the conflicts using MPC and firing other transition using ON/OFF strategy. The first two methods have very low computational complexity, while using the ON/OFF-MPC controller we may reach the final state faster, but with considerable higher computational complexity. Some comparisons are made by using different control methods and parameters.

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 strategies are proposed in Section 4. Section 5 compares these control methods by using an assembly system. Some conclusions are given in Section 6.
2. BASIC CONCEPTS AND NOTATIONS

2.1 Continuous Petri Nets

The reader is assumed to be familiar with basic concepts of continuous Petri nets (see [1, 9] for a gentle introduction).

**Definition 1.** A continuous PN system is a pair \((N, m_0)\) where \(N = (P, T, Pre, Post)\) is a net structure where:

- \(P\) and \(T\) are the sets of places and transitions respectively.
- \(Pre, Post \subseteq \mathbb{Q}_+^{[P] \times |T|}\) are the pre and post incidence matrices.
- \(m_0 \in \mathbb{R}_+^{[P]}\) is the initial marking (state).

For \(v \in P \cup T\), the sets of its input and output nodes are denoted as \(\bullet v\) and \(\bullet^* v\), respectively. Let \(p_i, i = 1, \ldots, |P|\) and \(t_j, j = 1, \ldots, |T|\) denote the places and transitions. Each place can contain a non-negative real number of tokens, its marking. The distribution of tokens in places is denoted by \(m\). The enabling degree of a transition \(t_j \in T\) is given by:

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

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

Non negative left and right natural annihilors of the token flow matrix \(C\) are called \(P\)-semiflows (denoted by \(y\)) and \(T\)-semiflows (denoted by \(x\)), respectively. If \(\exists y > 0\), \(y \cdot C = 0\), then the net is said to be conservative. If \(\exists x > 0\), \(C \cdot x = 0\) 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 occur from every reachable marking.

If \(m\) is reachable from \(m_0\) through a finite sequence \(\sigma\), the state (or fundamental) equation is satisfied:

\[m = m_0 + C \cdot \sigma, \quad \sigma \in \mathbb{R}_+^{[T]}\]

The firing count vector \(\sigma\) is said to be minimal if for any \(T\)-semiflow \(x\), \(||x|| \not\subseteq ||\sigma||\), where \(||\cdot||\) stands for the support of a vector, i.e., the index of the elements different than zero. A minimal firing count vector \(\sigma\), driving the system from \(m_0\) to \(m_f\) can be computed by solving the following linear programming problem (LPP):

\[
\begin{align*}
\min & \quad \mathbf{1}^T \cdot \sigma \\
\text{s.t.} & \quad m_f = m_0 + C \cdot \sigma \\
& \quad \sigma \geq 0
\end{align*}
\]

If for all \(p \in P\), \(|p^*| \leq 1\) then \(N\) is called Choice-Free PN (CFPN). A CFPN is structurally persistent in the sense that independently of the initial marking, the net has no conflict.

In timed continuous PN (TCPN) the state equation has an explicit dependence on time: \(m(\tau) = m_0 + C \cdot \sigma(\tau)\) which through time differentiation becomes \(\dot{m}(\tau) = C \cdot \dot{\sigma}(\tau)\). The derivative of the firing count \(f(\tau) = \dot{\sigma}(\tau)\) is called the firing flow. Depending on how the flow is defined, many firing server semantics appear, being the most used ones infinite (or variable speed) and finite (or constant speed) server semantics [1, 9], for which a firing rate \(\lambda_j \in \mathbb{R}_{>0}\) is associated to transition \(t_j\). This paper deals with infinite server semantics for which the flow of a transition \(t_j\) at time \(\tau\) is the product of its firing rate, \(\lambda_j\), and its enabling degree at \(m(\tau)\):

\[f(t_j, \tau) = \lambda_j \cdot \text{enab}(t_j, m(\tau)) = \lambda_j \cdot \min_{\rho \in \bullet^* t_j} \left\{ \frac{m(p_i, \tau)}{Pre(p_i, t_j)} \right\}\]

2.2 Control Problem

In this paper the net system is considered to be subject to external control actions, and it is assumed that the only admissible control law consists in slowing down the firing speed of transitions [9]. Under this assumption, the controlled flow of a TCPN system is denoted as: \(\dot{u}(\tau) = f(\tau) - u(\tau)\), with \(0 \leq u(\tau) \leq f(\tau)\). The overall behavior of the system is ruled by: \(\dot{m}(\tau) = C \cdot (f(\tau) - u(\tau))\). In this paper, it is assumed that every transition is controllable \(t_j\) is uncontrollable if the only control that can be applied is \(u(t_j) = 0\).

The control problem addressed here is to design a control action \(u\) that drives the system from the initial marking \(m_0\) to the desired final marking \(m_f\).

3. ON/OFF CONTROLLER AND ITS PROBLEMS

By sampling the continuous-time TCPN system with a sampling period \(\Theta\), we obtain the discrete-time TCPN [5] given by:

\[m_{k+1} = m_k + C \cdot w_k \cdot \Theta\]

\[0 \leq w_k \leq f_k\]

Here \(m_k\) and \(w_k\) are the marking and controlled flow at sampling instant \(k\), i.e., \(\tau = k \cdot \Theta\).

It is proved in [5] that if the sampling period satisfies (3), the reachability spaces of discrete-time and continuous-time PN systems are the same, excepting at borders.

\[
\forall p \in P: \sum_{t_j \in p^*} \lambda_j \cdot \Theta < 1
\]

In this paper, we assume that the sampling period \(\Theta\) satisfies (3).

An ON/OFF controller is proposed in [12] for structurally persistent PN, where every transition is fired as fast as possible at any time step until an upper bound, the minimal firing count vector \(\sigma\), is reached. Alg. 1 computes the firing flow of the ON/OFF controller for each time step. In the case of CFPN, using the ON/OFF controller based on the minimum firing count vector, the desired marking is reached in minimum time. However, in the case of general nets this is not true in general.

Let us notice that, different from CFPN, in a PN with general structure, its minimal firing count vector may be not unique for given initial and final states.

The main advantage of the ON/OFF control strategy is its low computational complexity. Given a (minimal)
Algorithm 1 ON/OFF controller
Input: $m_0, m_f, \sigma, C, \lambda, \Theta$
Output: $w_0, w_1, w_2, \ldots$
1: $k = 0$
2: while $\sum_{i=0}^{k-1} w_i \cdot \Theta \neq \sigma$ do
3: \text{Solve the following LPP:}
4: \begin{align*}
& \text{max } 1^T \cdot w_k \\
& \text{s.t. } m_{k+1} = m_k + C \cdot w_k \cdot \Theta \\
& 0 \leq w_k \cdot \Theta \leq \sigma - \sum_{i=0}^{k-1} w_i \cdot \Theta
\end{align*} 
5: \text{Apply } w_k : m_{k+1} = m_k + C \cdot w_k \cdot \Theta
6: \quad k := k + 1
7: \text{end while}
8: \text{return } w_0, w_1, w_2, \ldots

firing count vector (which can be computed in polynomial time), the control actions can be obtained by solving a simple LPP in each time step. But when the system is not CF, the convergence of the final state may not be ensured. Moreover, in the case of non-CFPN, conflicts ($|p^*| > 1$) may appear and by applying the ON/OFF strategy, firing one transition faster may reduce the firing of another transition, and the overall time for reaching $m_f$ may not be minimal. The following example shows a live and bounded system, but by applying the ON/OFF strategy, the final state is not reached.

\begin{example}
Assume we want to drive the system in Fig.1 to final state $m_f = [0.5 \ 0.5 \ 0.5 \ 0.5 \ 0.2 \ 0.4 \ 1.4]^T$, and the firing rate of $t_3$ is 100, while the firing rates of other transitions are all set to 1. $\sigma = [0.8 \ 1.3 \ 0.5 \ 0.1 \ 0 \ 0 \ 0]^T$ is a minimal firing count vector driving $m_0$ (shown in the figure) to $m_f$. Using this setting and applying the ON/OFF controller, $m_f$ is not reached. This is because $t_3$ is fired much faster than $t_2$ ($\lambda_3 \gg \lambda_2$), and consequently all the tokens in $p_3$ will go to $p_7$, leading $p_6$ to be emptied in the limit. If $p_6$ gets emptied, $t_5$ can not be fired, and $p_5$ is also emptied in the limit. Notice that, if $t_7$ could be fired, $p_6$ could get some tokens, but according to the control law, it is not allowed because $\sigma(t_7) = 0$.
\end{example}

![Fig. 1. A live and bounded CPN system](image)

### 4. EXTENDED ON/OFF CONTROLLER

In this section, the ON/OFF controller that cannot be directly applied to general CPN (see Ex.2 for an example when desired final marking is not reached with standard ON/OFF) is extended. Three extensions are proposed: ON/OFF+ controller, B-ON/OFF controller and ON/OFF-MPC controller.

#### 4.1 ON/OFF-plus (ON/OFF+) controller

The problem of the ON/OFF controller may arise from the incorrect manner of solving the conflicts (e.g., between $t_2$ and $t_3$ in Fig.1). Two transitions $t_a$ and $t_b$ are in a (structural) conflict relation if $t_a \cap t_b \neq \emptyset$. Here let us define the coupled conflict relation as its transitive closure.

In order to overcome this problem, we will force the flows of transitions that are in coupled conflict relation to be proportional to the given firing count vector, while for the other transitions the ON/OFF strategy is applied.

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

Algorithm 2 ON/OFF+ controller
Input: $m_0, m_f, \sigma, C, \lambda, \Theta$
Output: $w_0, w_1, w_2, \ldots$
1: $k = 0$
2: while $\sum_{i=0}^{k-1} w_i \cdot \Theta \neq \sigma$ do
3: \text{Solve the following LPP:}
4: \begin{align*}
& \text{max } 1^T \cdot w_k \\
& \text{s.t. } m_{k+1} = m_k + C \cdot w_k \cdot \Theta \\
& 0 \leq w_k \cdot \Theta \leq \sigma - \sum_{i=0}^{k-1} w_i \cdot \Theta
\end{align*} 
5: \text{Apply } w_k : m_{k+1} = m_k + C \cdot w_k \cdot \Theta
6: \quad k := k + 1
7: \text{end while}
8: \text{return } w_0, w_1, w_2, \ldots

The procedure of ON/OFF+ controller is similar to the one of standard ON/OFF, except the last constraint of LPP (5), which means that, in any time step $k$, if transitions $t_a$ and $t_b$ are in conflict, the following will be forced: $w_a(t_a) = \frac{\sigma(t_a)}{\sigma(t_b)} \cdot w_b(t_b)$. Notice that, only transitions with positive values in the corresponding firing count vector should be considered. In the following, it is assumed that transitions in a coupled conflict relation are enabled. Notice that this condition is rather weak since it is, for instance, verified by any system that can reach a positive marking.

In order to prove the convergence of Alg.2, it is first shown that the original system with the ON/OFF+ controller is equivalent to a CFPN system with a particular controller, i.e., the same state trajectory can be obtained. It is clear that $m_f$ is reached in the CFPN system, implying that it is also reached in the original one.

\begin{reduction}
Let $T_j = \{t_1, t_2, \ldots, t_n\}$ be a set of transitions of net $\mathcal{N} = (P, T, \text{Pre}, \text{Post})$ that are in coupled conflict relation. These transitions will be fired
proportionally according to a given firing count vector $\sigma$, i.e., for any $t_a, t_b \in T_j$, if $t_a$ is fired in an amount $s_a$, simultaneously, $t_b$ is fired in an amount $s_b$, such that $\frac{s_a}{s_b} = \frac{\sigma(t_a)}{\sigma(t_b)}$. Let $\bar{\sigma} = \sum_{t \in T} \sigma(t), N$ is transformed to $N' = \langle P', T', \text{Pre}', \text{Post}' \rangle$ 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. $\forall p \in \text{Pre}(p, t_j) = \sum_{t \in T} \text{Pre}(p, t) \cdot \sigma(t) / \bar{\sigma}$
4. $\forall p \in T_j$, $\text{Post}'(p, t_j) = \sum_{t \in T} \text{Post}(p, t) \cdot \sigma(t) / \bar{\sigma}$

**Example 3.** Let $m > 0$ and $\sigma(t_1) > 0, \sigma(t_2) > 0$. Fig. 2 shows how to merge two conflicting transitions $t_1$ and $t_2$ to $t_{1,2}$.

![Fig. 2. Reduction rule: merging $t_1$ and $t_2$](image)

**Proposition 4.** Let $S = \langle N, m_0 \rangle$, and $S' = \langle N', m_0 \rangle$ be the transformed system from $S$ by merging $T_j = \{ t_1, t_2, \ldots, t_n \}$ to $t_j$ by using the reduction rule. If in $S$, the transitions in $T_j$ are fired proportionally according to a given firing count vector $\sigma$, and in $S'$, transition $t_j$ is fired in an amount equal to the sum of the firing amounts of transitions in $T_j$, then the same marking is reached in $S$ and $S'$.

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

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

$$m_1(p_2) = m_0(p_2) - g_2 \cdot \alpha \cdot \sigma(t_1) - g_3 \cdot \alpha \cdot \sigma(t_2)$$

In the transformed system, if $t_{1,2}(s_1 + s_2)$ is fired, the new marking of $p_2$ is:

$$m_1'(p_2) = m_0(p_2) - (s_1 + s_2) \cdot \frac{g_2 \cdot \sigma(t_1) + g_3 \cdot \sigma(t_2)}{\sigma(t_1) + \sigma(t_2)}$$

$$= m_0(p_2) - \alpha \cdot \sigma(t_1) + \sigma(t_2) \cdot \frac{g_2 \cdot \sigma(t_1) + g_3 \cdot \sigma(t_2)}{\sigma(t_1) + \sigma(t_2)}$$

$$= m_1(p_2).$$

Similarly, for the places $p_1$ and $p_3$, they also follow the equality of the markings in both systems.

**Corollary 5.** If $m_f > 0$ is reachable in $S$ by firing $\sigma$ from $m_0 > 0$, then $m_f$ is reachable in $S'$ by firing $\bar{\sigma}'$, where:

$$\sigma'(t_j) = \sum_{t \in T} \sigma(t)$$

if $t_j$ is obtained by merging a set of transitions $T_j$

$$\sigma(t_j)$$

otherwise

**Proposition 6.** Let $S = \langle N, \lambda, \Theta, m_0 \rangle$ be a discrete-time TCPN system with $m_0 > 0$. Let $m_f > 0$ be a reachable final marking, such that $m_f = m_0 + C \cdot \sigma$. By applying the ON/OFF+ controller, $m_f$ is reached in finite time.

**Proof:** Let $S' = \langle N', \lambda', \Theta', m_0 \rangle$ be the system transformed from $S$ by merging all the conflicting transitions using the reduction rule (therefore $S'$ is CFPN).

Assume there exists a controller $A$ applied to $S'$, with $w'_f(t_j)$ the controlled flow in 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 $w'_f(t_j) = \sum_{t \in T_j} w_k(t_j)$; (2) otherwise $w'_f(t_j) = w_k(t_j)$, where $w_k(t)$ is flow of transition $t$ in $S$ that is controlled by using ON/OFF+ controller. Then, according to Proposition 4, the state trajectory of $S'$ obtained by applying controller $A$ is the same as in $S$ obtained by applying ON/OFF+ controller. Therefore it is equivalent to prove that by applying controller $A$ to $S'$, $m_f$ is reached in finite time.

This controller $A$ always exists, because if the firing rate of $t_j$ is big enough, case (1) can always be satisfied, using a positive control input $w_k(t_j)$. For case (2) we simply use the ON/OFF strategy and the same firing rate as in $S$.

Finally, let us notice that $S'$ is a CFPN, so for sure controller $A$ can drive $S'$ to its final state in finite time [12], implying that by applying ON/OFF+ controller to $S$, the final state is also reached.

**Remark 7.** The results of Proposition 6 can be naturally extended to continuous-time TCPN by taking sampling period $\Theta$ tending to 0.

It should be noticed that for continuous timed system 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 $m_f$, the final marking is reached at the limit, i.e., in infinite time. If $m_f > 0$ and using the proposed control method, this situation does not happen.

### 4.2 Balanced ON/OFF Controller (B-ON/OFF)

Using the ON/OFF+ controller, we can ensure that the final state $m_f > 0$ is reached from $m_0 > 0$ in finite time. But the main drawback of this method is also obvious. Since a set of transitions in coupled conflict relation are forced to be fired proportionally, the number of steps needed for firing $\sigma$ is decided by the slower ones. Therefore, in extreme cases, when some of these transitions have very small flows, the whole system may also be slowed down.

**Example 8.** Let us consider the simple (sub-)system in Fig.3, assuming that $t_1, t_2$ have the same firing rate equal to 1. Moreover, they are forced by a given $\sigma$ to fire in the same amounts. It is obvious that the flow of $t_2$ is 100 times faster than the flow of $t_1$, but if $t_1$ and $t_2$ should fire proportionally according to $\sigma$, $t_2$ is slowed down.

To overcome these extreme bad cases, first we fire the fast transitions and stop the slow ones for some time periods, expecting that the flows (speeds) of the slow transitions are increased, i.e., we will try to balance the fast and slow transitions. After that, the pure ON/OFF+ controller is applied until the final state is reached.
Fig. 3. Fast transitions may be slowed down

We will first show how to classify the slow and fast transitions, then this balancing strategy is presented.

Assume that the system is at marking \( m, w \) is the corresponding flow, and let \( \sigma \) be the firing count vector that should be fired to reach \( m_f \). Then \( s_j = \left[ \frac{\sigma(t_j)}{w(t_j)} \right] \) can be viewed as the estimation of number of steps that transition \( t_j \) needs to be fired. For two transitions \( t_a \) and \( t_b \), if \( s_a > s_b \), then \( t_a \) is slower than \( t_b \).

Let the estimation of the number of steps for transition \( t_j \) at \( m_0 \) be defined by:

\[
s_j^0 = \left[ \frac{\sigma(t_j)}{\lambda_j \cdot \text{enab}(t_j, m_0) \cdot \Theta} \right]
\]

If \( \text{enab}(t_j, m_0) = 0 \) then \( s_j^0 = \infty \).

Let us consider again the system in Ex.8 and let \( \sigma(t_1) = \sigma(t_2) = 10, \Theta = 0.01 \). The initial estimation of the steps number is: \( s_0^0 = 1000, s_2^0 = 10 \).

Based on this initial estimation, for any given set of transitions \( T_2 \) that are in coupled conflict relation, we will partition it into two subsets, the fast ones \( T_1 \) and the slow ones \( T_2 \), such that:

\[
\begin{align*}
T_1 \cap T_2 &= \emptyset, T_1 \cup T_2 = T_c \\
\forall t_a &\in T_1, t_b \in T_2, s_a^0 / s_b^0 > d \\
\forall t_{a1}, t_{a2} &\in T_1, 1/d \leq s_{a1}^0 / s_{a2}^0 \leq d
\end{align*}
\]

where \( d \geq 1 \) is a design parameter used to classify slow and fast transitions.

From (7), the estimations of number of steps that the transitions in \( T_2 \) need to fire, are at least \( d \) times greater than the ones of transitions in \( T_1 \). If we fire the transitions in \( T_1 \) and \( T_2 \) proportionally, transitions in \( T_1 \) are obviously slowed down by the ones \( T_2 \).

Notice that, if the value of \( d \) is too big, all the transitions are put into \( T_1 \), then it is equivalent to applying ON/OFF+ controller directly. On the other side, if \( d \) is too small, then all the transitions are put into \( T_2 \), then they are all stopped.

In the system shown in Ex.8, 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 in \( k^{th} \) time step at \( m_{k-1} \), and the firing count vector that has been fired \( \sigma' \), i.e., \( m_{k} = m_{k-1} + C \cdot \sigma' \). The remained firing count vector should be fired is \( \sigma_k = \sigma - \sigma' \geq 0 \). The estimation of number of steps for transition \( t_j \in T_c \) at \( m_k \) is defined by:

\[
s_j^k = \begin{cases} 
\left[ \frac{\sigma(t_j)}{w(t_j)} \right], & \text{if } t_j \in T_1 \\
\left[ \frac{\sigma(t_j)}{\lambda_j \cdot \text{enab}(t_j, m_{k-1})} \right], & \text{if } t_j \in T_2
\end{cases}
\]

where \( w_k(t_j) \) is the flow of transition \( t_j \) when the ON/OFF+ strategy is applied. Notice that, since the transitions in \( T_1 \) are fired proportionally, for any \( t_j \in T_1 \), the same estimation \( s_j^k \) is obtained, denoted by \( h_k \).

For any \( t_0 \in T_2 \), let \( D_b^k = s_b^k / h_k \), it reflects the difference of the estimations between \( t_0 \) and the faster transitions.

Let \( T_p \) be the set of persistent transitions (those transitions that are not in a coupled conflict relation), and \( T^i_p, i = 1, 2, 3, \ldots, l \) be the sets of transitions in coupled conflict. Alg. 3 gives the control method: for transitions in \( T_p \), ON/OFF strategy is always applied; for any \( T^i_p \), \( T^i_p \) are fired proportionally using ON/OFF+ strategy, while every slow transition \( t_0 \) in \( T^i_p \) is stopped until the following condition (C1) or (C2) is satisfied, then it is moved to \( T^i_1 \) and also fired using ON/OFF+ strategy.

(C1) \( D_b^k \leq d \).

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

By stopping \( t_0 \) while firing other transitions, tokens may be put into the input places of \( t_0 \), consequently increasing its flow, then \( t_0 \) may become more balanced with those fast transitions, i.e., \( D_b^k \) is decreased. If \( D_b^k \) is keeping decreased, for sure in finite time, we will have condition (C1) satisfied, it means that \( t_0 \) is already balanced with the fast transitions. If at one moment, \( D_b^k \) can not be decreased anymore, then condition (C2) is satisfied, i.e., transition \( t_0 \) can not become more balanced with the fast ones. Therefore, one of this conditions will be satisfied in finite time. After that, there is no need to stop \( t_0 \) and we should start to fire it using the pure ON/OFF+ strategy.

Now we will prove the convergence of this B-ON/OFF controller, i.e., the final state is reached in finite time.

**Proposition 9.** Let \((N, \lambda, m_0)\) be a TCPN system, with \( m_0 > 0 \). Let \( m_f > 0 \) be a reachable final marking, such that \( m_f = m_0 + C \cdot \sigma \). By applying the B-ON/OFF controller, \( m_f \) is reached in finite time.

**Proof:** For all the slow 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 \( m > 0 \) and \( m_f \) is reachable from \( m \).

Since the accumulative firing counts of transitions are upper bounded by \( \sigma \), then we have \( m = m_0 + C \cdot \sigma' \), \( 0 \leq \sigma' \leq \sigma \). Since \( \sigma - \sigma' \geq 0 \) and \( m_f = m + C \cdot (\sigma - \sigma') \), \( m > 0 \), \( m_f \) is reachable from \( m \) [3].

### 4.3 ON/OFF-MPC controller

MPC for TCPN is proposed in [5], where in each time step the following optimization problem is solved:

\[
\begin{align*}
\min & \quad J(m_k, N) \\
\text{s.t.} & \quad m_{k+j+1} = m_{k+j} + \Theta \cdot C \cdot w_{k+j}, \quad (8a) \\
& \quad G \cdot [w_{k+j}] \leq 0, j = 0, ..., N - 1 \quad (8b) \\
& \quad w_{k+j} \geq 0, j = 0, ..., N - 1 \quad (8c)
\end{align*}
\]
For example, in the quadratic case, net structure and (8b) gives the constraint on firing flows. Let us denote by controller, but with higher computational complexity. If the final state/input is not an interior point, by forcing reach the desired state as soon as possible, i.e., minimizing The problem that should be solved in each time step along is shown in Alg.4.

MPC is usually used for optimizing trajectories satisfying certain objective functions. In our problem, the aim is to reach the desired state as soon as possible, i.e., minimizing the time. Even if it is difficult to obtain a minimum time control by using an MPC approach, we will consider this method for transitions in conflicts while for the others we will keep the ON/OFF controller. We will show that in some situations the number of steps to reach the desired final state is smaller than for ON/OFF+ or B-ON/OFF controller, but 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 ON/OFF-MPC controller is shown in Alg.4.

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

$$
\min \ J(m_k, N) \\
n.s.t.: \ m_{k+j+1} = m_{k+j} + C \cdot w_{k+j} \cdot \Theta, \quad (10a)
$$

$$
G \cdot \begin{bmatrix} w_{k+j} \\ m_{k+j} \end{bmatrix} \leq 0, \quad j = 0, ..., N - 1 \quad (10b)
$$

$$
w_{k+j} \geq 0, \quad j = 0, ..., N - 1 \quad (10c)
$$

$$
w_{k+j} \leq (\sigma(t_d)/\Theta) \forall t_d \in T_c \quad (10d)
$$

where $f_k(t_d)$ is the uncontrolled flow of transition $t_d$ at $m_k$. (10d) gives the upper bound of accumulative firing counts and (10e) makes sure that if a transition is persistent, it is fired using the ON/OFF strategy. As defined in the unconstrained Linear Quadratic Regulation (LQR), let $K, P$ be the solution of (11) (see [8]), and let $Z = P$. Using results from the classical optimal control theory, we can guarantee the convergence to the desired state only if the set of feasible state and input vectors are bounded and the final state and input are interior points. If the final state/input is not an interior point, by forcing straight line trajectories, the asymptotic stability can be also achieved [5].

$$
K = -(R + B^T P B)^{-1} B^T P A \\
P = (A + BK)^T P (A + BK) + K^T R K + Q
$$

5. SIMULATIONS

In this section, the previous control methods are applied to a CPN model of an assembly system. The simulations are performed on a PC with Intel(R) Core(TM)2 Quad CPU Q940 2.66GHz, 3.24GB of RAM. The system model in Fig. 4 represents an assembly system. There are two kinds of input raw materials stored in $p_1$ and $p_2$. The material A, B are first processed by $\text{Proc}_A1$, then the obtained semi-products are further processed by $\text{Proc}_A2$ and $\text{Proc}_A3$. In the other processing line, material B is sequentially processed by $\text{Proc}_B1$ and $\text{Proc}_B2$. Then final produces are obtained after assembling all the semi-products.

It is assumed that the firing rate of $t_2$ is 4, while for the other transitions, are equal to 1. The simulations are performed under different setting, case:

1) $\Theta = 0.01, \ m_0 = [1 \ 2 \ 0.4 \ 0.5 \ 0 \ 0 \ 0.5 \ 0]^T, \ \sigma = [0.4 \ 0.2 \ 0.5 \ 0.3 \ 0.1 \ 0]^T, \ m_f = [0.6 \ 1.8 \ 0.7 \ 0.2 \ 0.1 \ 0.4 \ 0.2 \ 4.7 \ 0.3]^T$;
2) $\Theta = 0.01, \ m_0 = [1 \ 2 \ 0.001 \ 0.5 \ 0 \ 0 \ 0 \ 0]^T, \ \sigma = [0.4 \ 0.2 \ 0.5 \ 0.3 \ 0.1 \ 0]^T, \ m_f = [0.6 \ 1.8 \ 0.301 \ 0.2 \ 0.1 \ 0.4 \ 0.2 \ 4.7 \ 0.3]^T$;
Fig. 4. The TCPN model of an assembly system.

Table 1 gives the number of time steps required for reaching $m_f$ from $m_i$ and the CPU times used for computing the corresponding control laws. The results of MPC are conducted using matrices $Q = 1000 \cdot I_n$, $R = 0.01 \cdot I_n$, and $Z = P$, with $P$ solution of (11). In the B-ON/OFF controller, $d$ is set to 10.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Time steps</th>
<th>CPU time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting</td>
<td>1) 2) 3)</td>
<td>1) 2) 3)</td>
</tr>
<tr>
<td>ON/OFF+</td>
<td>124</td>
<td>954 85</td>
</tr>
<tr>
<td>ON/OFF</td>
<td>124</td>
<td>259 85</td>
</tr>
<tr>
<td>ON/OFF-MPC (N=1)</td>
<td>120</td>
<td>158 203</td>
</tr>
<tr>
<td>ON/OFF-MPC (N=20)</td>
<td>118</td>
<td>149 90</td>
</tr>
</tbody>
</table>

Table 1. Simulation Results

From the simulation results, it can be observed that the first two ON/OFF strategies have very low computational complexity. The B-ON/OFF controller does not improve the number of time steps in case 1) and 3), because the flows of conflicting transitions are similar, so it may make sense to directly fire them proportionally. But in the case 2), the B-ON/OFF controller is much better. As for the ON/OFF-MPC controller, the computational complexity is not very high when $N = 1$. Its number of time steps can be improved when greater horizon step is used, but consequently, the computing time is significantly increased. Finally, let us observe that, in this example, we may have “deadlock” when applying the standard ON/OFF controller.

Notice that we have shown the results of different methods for a particular example, but it does not indicate one method is definitely better than another in a general sense.

6. CONCLUSIONS

In this work, three ON/OFF strategy based extensions are presented. It is proved that they can drive a general CPN system to the desired final state in finite time. Some comparisons are also given. The advantage of these ON/OFF based controllers is the low computational complexity. As a future work, we will compare our methods with other control strategies, and consider how to identify the most suitable controller in different situations.

REFERENCES