



HAL
open science

Event-Driven Control for Reaching a Steady State in Controlled Generalized Batches Petri Nets

Ruotian Liu, Rabah Ammour, Leonardo Brenner, Isabel Demongodin

► **To cite this version:**

Ruotian Liu, Rabah Ammour, Leonardo Brenner, Isabel Demongodin. Event-Driven Control for Reaching a Steady State in Controlled Generalized Batches Petri Nets. 15th IFAC Workshop on Discrete Event Systems, WODES 2020, Nov 2020, Rio de Janeiro, Brazil. 10.1016/j.ifacol.2021.04.063 . hal-03171923

HAL Id: hal-03171923

<https://amu.hal.science/hal-03171923v1>

Submitted on 27 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Event-driven control for reaching a steady state in controlled Generalized Batches Petri Nets

Ruotian Liu, Rabah Ammour, Leonardo Brenner,
Isabel Demongodin

*Aix-Marseille University, University of Toulon, CNRS, LIS, Marseille,
France (e-mail: {ruotian.liu, rabah.ammour, leonardo.brenner,
isabel.demongodin}@lis-lab.fr).*

Abstract: This paper addresses the control problem of controlled generalized batches Petri nets. The considered problem is to reach a given steady state from an initial one by controlling the firing flow of transitions. The proposed strategy is event-driven and the control actions are updated only when specific events occur. It is based on an algorithm with a linear programming problem that computes a new control firing flow vector when the considered events occur. The convergence of the marking to its steady state marking by the proposed strategy is proved and an example is provided to illustrate the computation steps.

Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0>)

Keywords: Discrete event systems, Petri nets, event-driven control, hybrid systems

1. INTRODUCTION

Generalized batches Petri net (GBPN) (Demongodin, 2001) enriches the class of hybrid Petri nets (David and Alla, 2005) by introducing a new kind of nodes, called batch nodes. Batch transitions act as continuous ones with finite server semantics while batch places are hybrid ones defined by three continuous characteristics: a length, a transfer speed and a maximal density. Based on the concept of batches as marking, i.e., a group of entities moves through a batch place at a certain speed, GBPNs allow variable delays on continuous flows to be represented. Behavior of such hybrid models is based on a switching continuous-time and event-driven dynamics. When the firing flows of continuous and batch transitions and the transfer speeds of batch places are considered as control inputs, controlled GBPNs (cGBPNs) have been defined (Demongodin and Giua, 2012).

One problem in dynamical systems is that of determining a stationary state, which can be maintained for an indefinite time. In a cGBPN without discrete nodes, this stationary state has been defined as a steady state (Demongodin and Giua, 2012) where the markings of places and the instantaneous firing flows of transitions remain constant. The existence and choice of an optimal steady state has been proposed in Demongodin and Giua (2014).

In this work, at first, we assume such a steady state is determined from an initial marking. Our objective is to reach this steady state from the initial one by controlling the firing flows of continuous and batch transitions. Some important works have been devoted to these control problems with continuous Petri nets under infinite server semantics, applying discrete-time control such as

ON/OFF controllers (Wang et al., 2014) or Model Predictive Control (Mahulea et al., 2008). As stated by Casandras (2014), the event-driven control strategies provide an alternative to time-driven ones in hybrid systems where events decompose state trajectories into different discrete modes. In this spirit, Júlvez et al. (2014) have proposed an event-driven model predictive control of timed hybrid Petri nets. In our work, dedicated to cGBPNs, the proposed control strategy is a pure event-driven one where the control actions are updated only when specific events occur and remain constant between timed events.

Based on the assumptions that the considered cGBPN is conservative, consistent and without discrete nodes, the control strategy based on an ON/OFF controller, assumes that the controlled firing flow of each transition cannot exceed its steady firing flow. This is motivated by the phenomenon in many physical systems where the applied control values are limited by their nominal ones rather than the maximal values. Moreover, a certain configuration of the moving entities characterized by a steady state density should be reached and the usage of the steady firing flow allows such configuration to be generated.

The content of this paper is structured as follows. In Section 2, we recall the basic definitions on controlled generalized batches Petri nets (cGBPN) and steady state of such models. Section 3 formalizes the control problem and the related assumptions. In Section 4, we address the problem of the event-driven control strategy, the developed algorithm is detailed and its convergence is proved by analyzing the transient behavior of the controlled net. Finally, some conclusions and future works are presented in Section 5.

¹ This work has been partially supported by the CPSecurity project (CNRS-INS2I grant).

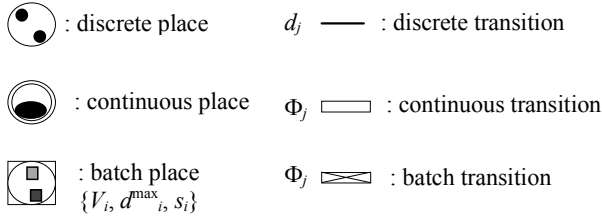


Fig. 1. Nodes of a GBPN.

2. BASIC CONCEPTS AND NOTATIONS

This section introduces some basic definitions on controlled generalized batches Petri nets (cGBPN) and characterizes the steady states.

2.1 Generalized Batches Petri nets

We assume the reader to be familiar with hybrid Petri nets such as defined by David and Alla (2005). For more details on batches Petri net formalisms, the reader can refer to (Demongodin, 2001) and (Demongodin and Giua, 2014).

Definition 1. A *Generalized Batches Petri net* (GBPN) is a 6-tuple $N = (P, T, \text{Pre}, \text{Post}, \gamma, \text{Time})$ where:

- $P = P^D \cup P^C \cup P^B$ is finite set of places partitioned into the three classes of *discrete*, *continuous* and *batch* places.
- $T = T^D \cup T^C \cup T^B$ is finite set of transitions partitioned into the three classes of *discrete*, *continuous* and *batch* transitions.
- $\text{Pre}, \text{Post} : (P^D \times T \rightarrow \mathbb{N}) \cup ((P^C \cup P^B) \times T \rightarrow \mathbb{R}_{\geq 0})$ are², respectively, the pre-incidence and post-incidence matrices, denoting the weight of the arcs from places to transitions and transitions to places.
- $\gamma : P^B \rightarrow \mathbb{R}_{>0}^3$ is the *batch place function*. It associates to each batch place $p_i \in P^B$ the triple $\gamma(p_i) = (V_i, d_i^{\max}, s_i)$ that represents, respectively, *maximal transfer speed*, *maximal density* and *length* of p_i .
- $\text{Time} : T \rightarrow \mathbb{R}_{\geq 0}$ associates a nonnegative number to every transition:
 - if $t_j \in T^D$, then $\text{Time}(t_j) = d_j$ denotes the *firing delay* associated with the discrete transition;
 - if $t_j \in T^C \cup T^B$, then $\text{Time}(t_j) = \Phi_j$ denotes the *maximal firing flow* associated with the continuous or batch transition. ■

The cardinality of the sets of places and transitions is denoted as $m^Y = |P^Y|$ and $n^Y = |T^Y|$, respectively, with $Y \in \{D, C, B\}$. Figure 1 shows the graphical representation of nodes of a GBPN.

The maximal capacity of batch place p_i , with $\gamma(p_i) = (V_i, d_i^{\max}, s_i)$, is $Q_i = s_i d_i^{\max}$. To ensure a structural bound on a batch place p_i with capacity Q_i , a complementary continuous place p_j must exist such that $\text{Pre}(p_i, \cdot) = \text{Post}(p_j, \cdot)$ and $\text{Post}(p_i, \cdot) = \text{Pre}(p_j, \cdot)$. Consequently $|P^C| \geq |P^B|$.

Definition 2. A batch β_k at time τ is defined by a triple $\beta_k(\tau) = (l_k(\tau), d_k(\tau), x_k(\tau))$, where $l_k(\tau) \in \mathbb{R}_{\geq 0}$ is the length, $d_k(\tau) \in \mathbb{R}_{\geq 0}$ is the density and, $x_k(\tau) \in \mathbb{R}_{\geq 0}$ is the head position. ■

The marking $\mathbf{m}(\tau) = [m_1(\tau) m_2(\tau) \dots m_m(\tau)]^T$ of a GBPN at time τ , is a function that assigns to each discrete place a nonnegative integer, to each continuous place a nonnegative real number and assigns to each batch place p_i , a series of batches ordered by their head positions, $m_i(\tau) = \{\beta_1(\tau), \dots, \beta_r(\tau)\}$. We denote by \mathbf{m}_0 the initial marking. Time τ will be omitted in the rest of paper when there is no ambiguity.

Another important notion in GBPNs is the definition of the marking quantity which, for a continuous or a discrete place, is equal to its marking while for a batch place is the sum of the quantities within the batches it contains.

Definition 3. The *marking quantity vector* $\mathbf{q} = \mu(\mathbf{m}) \in \mathbb{R}_{\geq 0}^m$ associated with a marking \mathbf{m} is defined as follows:

$$q_i = \begin{cases} m_i & \text{if } p_i \in P^D \cup P^C, \\ \sum_{\beta_k \in m_i} l_k \cdot d_k & \text{if } p_i \in P^B. \end{cases}$$

Remark 1. $\mu(\mathbf{m})$ is an injective mapping which associates to a given marking \mathbf{m} a single marking quantity vector \mathbf{q} . Its inverse is not, since more than one marking \mathbf{m} may correspond to a given marking quantity vector \mathbf{q} .

Every batch place, $p_i \in P^B$, is associated with an output density, denoted d_i^{out} , representing the density of an output batch of the place. Formally, if batch place p_i contains an output batch $\beta_r(\tau)$ at time τ , (i.e., $x_r(\tau) = s_i$), then $d_i^{\text{out}}(\tau) = d_r(\tau)$, else $d_i^{\text{out}}(\tau) = 0$.

The dynamics of a GBPN, ruled by the firing of enabled transitions and the hybrid dynamics inside batch places, uses the notion of instantaneous firing flow vector. More precisely, to a continuous or batch transition, $t_j \in T^C \cup T^B$, is associated an *instantaneous firing flow* (IFF) $\varphi_j(\tau) \leq \Phi_j$, that represents the quantity of firing of transition t_j by time unit. The IFF vector at time τ is denoted by $\varphi(\tau) \in \mathbb{R}^{n^C + n^B}$. The *input* (resp., *output*) flow of a batch or continuous place p_i at time τ is the sum of all flows entering (resp., leaving) the place and can be written, respectively, as: $\phi_i^{\text{in}}(\tau) = \text{Post}(p_i, \cdot) \cdot \varphi(\tau)$ and $\phi_i^{\text{out}}(\tau) = \text{Pre}(p_i, \cdot) \cdot \varphi(\tau)$.

The behavior of a GBPN is based on a timed discrete event dynamics with linear or constant continuous evolutions between timed events. Between two timed events, the state of the net is characterized by an invariant behavior state (IB-state) (Demongodin and Giua (2010)), which corresponds to a period of time such that the marking in discrete places is constant, the firing flow of continuous and batch transitions is constant and the output density of batch places is constant. Note that, as φ is constant in an IB-state, ϕ^{in} and ϕ^{out} are also constants between timed events. At time τ (i.e., date of an event), the set of admissible firing flows form a convex set described by linear equations (see Definition 3.1 in Demongodin and Giua (2010)). In particular, the firing flow of continuous and batch transitions, and the input/output flow of batch places must satisfy the following constraints: the firing flow

² We denote $\mathbb{R}_{\geq 0}$ (resp., $\mathbb{R}_{>0}$) the set of nonnegative (resp., positive) real numbers and \mathbb{N} the set of natural numbers.

of a transition that is not enabled is null; the total flow entering batch place p_i should not be greater than the maximal flow $V_i \cdot d_i^{\max}$ that the place can accept; the total flow exiting batch place p_i should not be greater than the exit flow $V_i \cdot d_i^{\text{out}}$ generated by the output batch of the place. Finally, the fundamental equation of cGBPNs is expressed by:

$$\mathbf{q}(\tau) = \mathbf{q}(\tau_0) + \mathbf{C} \cdot \mathbf{z}(\tau) \quad (1)$$

where $\mathbf{z}(\tau) \in \mathbb{R}_{\geq 0}^n$, called characteristic vector, denotes the firing count of each transition during time interval $[\tau_0, \tau]$, and $\mathbf{C} = \text{Post} - \text{Pre}$.

2.2 Controlled GBPN and steady state

When the instantaneous firing flow of continuous and batch transitions and the transfer speed of batch places are considered as control inputs, this leads to the definition of controlled GBPN.

Definition 4. A controlled Generalized Batches Petri net (cGBPN) is a GBPN N for which a control input $\mathbf{u}(\tau) = (\varphi(\tau), \mathbf{v}(\tau))$ is defined. ■

In this paper, we assume that the transfer speed of each batch place is constant, i.e., $v_i(\tau) = V_i$ for all $p_i \in P^B$. The control input is consequently restricted to the instantaneous firing flow vector, i.e., $\mathbf{u}(\tau) = \varphi(\tau)$ with $0 \leq \mathbf{u}(\tau) \leq \Phi$. The following definition, taken from Demongodin and Giua (2014), slightly restricts the definition of steady states when the speeds of batch places are assigned.

Definition 5. (Steady state) Let $\langle N, \mathbf{m}_0 \rangle$ be a cGBPN with $P^D = T^D = \emptyset$. The net is in a steady state at time τ_s if for $\tau \geq \tau_s$ the marking \mathbf{m}^s and the instantaneous firing flow vector φ^s remain constant. Thus a steady state is defined by a pair $(\mathbf{m}^s, \varphi^s)$. ■

Note that this definition also implies that the output density of batch places is constant at the steady state.

In Demongodin and Giua (2014)(see Proposition 4.5), we characterize the steady state of a cGBPN by solving a linear programming problem that takes into account the net structure and the initial marking. It has also been proved, that at the steady state, the marking has one of the following regular forms.

Proposition 6. Assume that a net $\langle N, \mathbf{m}_0 \rangle$ with $P^D = T^D = \emptyset$ is in a steady state $(\mathbf{m}^s, \varphi^s)$. The marking m_i^s of a batch place $p_i \in P^B$ – with input/output flow ϕ_i^s and marking quantity $q_i^s = \mu(m_i^s)$ – takes the following regular form:

- (1) If $\phi_i^s = 0$, marking $m_i^s = \{\beta_o\}$ contains a single dense output batch $\beta_o = (l_o, d_i^{\max}, s_i)$ of length $l_o = q_i^s / d_i^{\max}$.
- (2) If $q_i^s = \phi_i^s s_i / V_i > 0$, marking $m_i^s = \{\beta_o\}$ contains a single output batch $\beta_o = (s_i, d_o, s_i)$ of length equal to the length of the place and with density $d_o = \phi_i^s / V_i$.
- (3) If $Q_i > q_i^s > \phi_i^s s_i / V_i > 0$, marking $m_i^s = \{\beta_e, \beta_o\}$ contains a dense output batch $\beta_o = (l_o, d_i^{\max}, s_i)$ in contact with one input batch $\beta_e = (l_e, d_e, l_e)$ such that $d_e = \phi_i^s / V_i$ and

$$l_e = \frac{s_i d_i^{\max} V_i - q_i^s V_i}{d_i^{\max} V_i - \phi_i^s} \quad \text{and} \quad l_o = s_i - l_e.$$

- (4) If $Q_i = q_i^s > \phi_i^s s_i / V_i > 0$, marking $m_i^s = \{\beta_o\}$ contains a single dense output batch $\beta_o = (s_i, d_i^{\max}, s_i)$ in accumulation behavior of length equal to the length of the place.

Thus from \mathbf{q}^s and φ^s the regular marking \mathbf{m}^s can be uniquely reconstructed. ◊

The following example shows a cGBPN with its steady state.

Example 7. Let us consider the cGBPN shown in Fig. 2 where the characteristics of batch places are represented and the maximal firing flow vector is $\Phi = [4 \ 4 \ 4 \ 4 \ 4]^T$. Let the initial marking $\mathbf{m}_0 = [3 \ 0.4 \ 3 \ 0 \ \{(0.2, 3, 1)\} \ 0]^T$, that implies the initial quantity vector equal to $\mathbf{q}^0 = [3 \ 0.4 \ 3 \ 0 \ 0.6 \ 0]^T$.

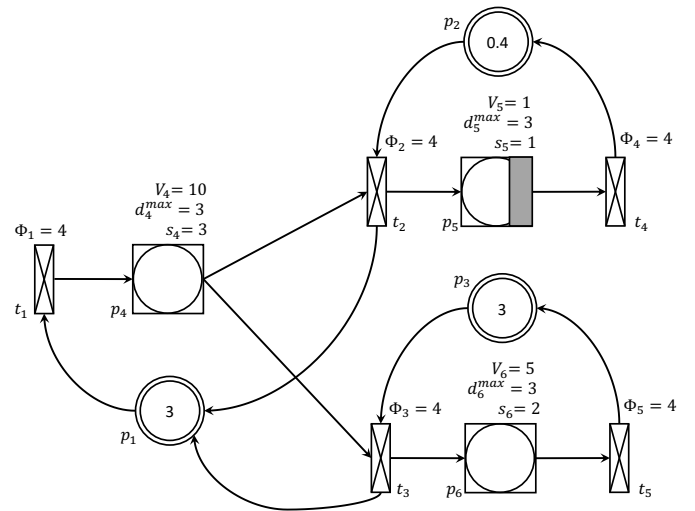


Fig. 2. The cGBPN of Example 7.

By optimizing the same objective function $f = 10 \cdot \varphi_4 + \varphi_5 - 10^{-6}(z_1 + z_2 + z_3 + z_4 + z_5)$ in Demongodin and Giua (2014), an optimal solution $(\mathbf{q}^s, \varphi^s)$ and the characteristic vector \mathbf{z}^s are obtained as: $\mathbf{q}^s = [1.8 \ 0 \ 1.8 \ 1.2 \ 1 \ 1.2]^T$, $\varphi^s = [4 \ 1 \ 3 \ 1 \ 3]^T$, $\mathbf{z}^s = [2.8 \ 0.4 \ 1.2 \ 0 \ 0]^T$.

From this optimal solution, and by Proposition 6, the reachable steady state $(\mathbf{m}^s, \varphi^s)$ is given by: $\mathbf{m}^s = [1.8 \ 0 \ 1.8 \ \{(3, 0.4, 3)\} \ \{(1, 1, 1)\} \ \{(2, 0.6, 2)\}]^T$ and $\varphi^s = [4 \ 1 \ 3 \ 1 \ 3]^T$.

3. PROBLEM STATEMENT

This paper addresses the problem of controlling the transient behavior of a cGBPN for reaching a given steady state from an initial marking. In particular, the problem is how to reach a target steady state $(\mathbf{m}^s, \varphi^s)$ from a given initial state \mathbf{m}_0 , by controlling only the instantaneous firing flow of transitions. Inspired by Aguayo-Lara et al. (2012), the control problem is formally defined as follows.

Definition 8. Let $\langle N, \mathbf{m}_0 \rangle$ be a cGBPN with $P^D = T^D = \emptyset$ and $(\mathbf{m}^s, \varphi^s)$ a reachable steady state. The event-driven control problem for reaching $(\mathbf{m}^s, \varphi^s)$ deals with

the computation of a timed control trajectory $(\mathbf{u}^0, \tau_0), (\mathbf{u}^1, \tau_1), \dots, (\mathbf{u}^i, \tau_i), \dots, (\mathbf{u}^s, \tau_s)$ feasible in the cGBPN from \mathbf{m}_0 such that $\mathbf{m}(\tau_f) = \mathbf{m}^s$ and $\mathbf{u}(\tau_f) = \mathbf{u}^s = \boldsymbol{\varphi}^s, \forall \tau_f \geq \tau_s$. ■

Note that when a cGBPN is without discrete nodes, the fundamental equation Eq.(1) becomes:

$$\mathbf{q}(\tau) = \mathbf{q}(\tau_0) + \mathbf{C} \cdot \int_{\tau_0}^{\tau} \mathbf{u}(\rho) d\rho \quad (2)$$

where $\int_{\tau_0}^{\tau} \mathbf{u}(\rho) d\rho$ denotes the sum of firing quantity of continuous or batch transition during time interval $[\tau_0, \tau]$. Consequently, for the control trajectory $(\mathbf{u}^0, \tau_0), (\mathbf{u}^1, \tau_1), \dots, (\mathbf{u}^i, \tau_i), \dots, (\mathbf{u}^s, \tau_s)$ such that the controlled firing flow vector \mathbf{u}^i is applied at date τ_i and remains constant until τ_{i+1} , corresponding to a timed interval of an IB-state, Eq. (2) becomes:

$$\mathbf{q}(\tau_s) = \mathbf{q}^0 + \mathbf{C} \cdot \left(\int_{\tau_0}^{\tau_1} \mathbf{u}^0 \cdot d\rho + \dots + \int_{\tau_{s-1}}^{\tau_s} \mathbf{u}^{s-1} \cdot d\rho \right). \quad (3)$$

In the rest of the paper, we consider the following assumptions under investigation.

- A1 No discrete nodes ($P^D = T^D = \emptyset$).
- A2 The steady firing flow vector is positive ($\boldsymbol{\varphi}^s > \mathbf{0}$).
- A3 The net is conservative.

Assumption (A1) preserves the restricted class of cGBPNs for which the steady state analysis has been studied (see Demongodin and Giua (2014) for more details). Assumption (A2) ensures that the net is consistent, i.e., there exists a positive T-semiflow. Assumption (A3) imposes that each place is contained in the support of a P-semiflow. As previously mentioned, each batch place is associated to a continuous one for respecting its structural bound. Thus it is sufficient to state that all continuous places are in a conservative component to ensure that the net is conservative. More details on properties of autonomous continuous Petri nets can be found in Júlvez et al. (2003) and Navarro-Gutiérrez et al. (2017).

4. THE PROPOSED EVENT-DRIVEN CONTROL STRATEGY

In this part, we propose an approach for the computation of a control trajectory from an initial state to the given steady state. The basic idea is to set the firing flow of an enabled transition t_j to zero when the marking quantity of one of its input places is lower than the steady state value and, for the other enabled transitions, to maximize their firing flows to the steady state ones for approaching the steady marking quantity vector \mathbf{q}^s . We assume that the controlled firing flow of each transition t_j cannot exceed the steady firing flow φ_j^s . The proposed control strategy could be summarized as follows³, at time τ .

- OFF: $u_j(\tau) = 0$ if t_j is not enabled or if t_j is enabled with $\exists p_i \in \bullet t_j : q_i(\tau) < q_i^s$.
- ON: maximize $\mathbf{u}(\tau)$ to $\boldsymbol{\varphi}^s$.

³ For $t \in T, \bullet t = \{p \in P \mid \text{Pre}(p, t) > 0\}$, and $t^\bullet = \{p \in P \mid \text{Post}(p, t) > 0\}$.

The controlled flow values are maintained during an IB-state and updated when a particular event (or possibly several events at the same time) occurs. The events considered by the controller are:

- The marking quantity of place p_i reaches its steady quantity q_i^s ;
- The output density of a batch place changes its value.

4.1 Computation of a timed control trajectory

Before we present the algorithm that allows one to compute a control trajectory, several sets must be defined at any marking \mathbf{m} .

- $S_L(\mathbf{m}) = \{p_i \in P \mid q_i < q_i^s\}$ be the subset of places whose marking quantities are lower than their steady state values.
- $S_G(\mathbf{m}) = \{p_i \in P \mid q_i > q_i^s\}$ be the subset of places whose marking quantities are greater than their steady state values.
- $S_E(\mathbf{m}) = \{p_i \in P \mid q_i = q_i^s\}$ be the subset of places whose marking quantities are equal to their steady state values.
- $T_N(\mathbf{m})$ be the subset of transitions that are not enabled at \mathbf{m} .
- $T_L(\mathbf{m}) = \{t_j \in T \mid \exists p_i \in \{S_L \cap \bullet t_j\}\}$ be the subset of transitions with at least one of its input places belonging to S_L .

The following algorithm computes a timed control trajectory from an initial state to a given reachable steady state.

Algorithm 1: Computation of control trajectory

Input: A cGBPN $\langle N, \mathbf{m}_0 \rangle$ and a reachable steady state $(\mathbf{m}^s, \boldsymbol{\varphi}^s)$.

Output: Control trajectory $(\mathbf{u}^0, \tau_0), (\mathbf{u}^1, \tau_1), \dots$

- 1 Initialize : $\mathbf{q}^0 = \mu(\mathbf{m}_0)$, $\mathbf{m}^0 = \mathbf{m}_0$, $\mathbf{u}^0 = \mathbf{0}$, $\tau_0 = 0$, $i = 0$;
- 2 **while** $\mathbf{m}^i \neq \mathbf{m}^s$ **do**
- 3 Determine $T_N(\mathbf{m}^i)$, $T_L(\mathbf{m}^i)$, $S_G(\mathbf{m}^i)$, $S_E(\mathbf{m}^i)$;
- 4 Solve the following LPP: $\max \mathbf{1}^T \cdot \mathbf{u}^i$ s.t.

$$\begin{cases} \text{(a)} & 0 \leq u_j^i \leq \varphi_j^s & \forall t_j \in T \\ \text{(b)} & u_j^i = 0 & \forall t_j \in T_N(\mathbf{m}^i) \cup T_L(\mathbf{m}^i) \\ \text{(c)} & C(p_k, \cdot) \cdot \mathbf{u}^i \leq 0 & \forall p_k \in S_G(\mathbf{m}^i) \\ \text{(d)} & C(p_k, \cdot) \cdot \mathbf{u}^i = 0 & \forall p_k \in S_E(\mathbf{m}^i) \\ \text{(e)} & \text{Pre}(p_k, \cdot) \cdot \mathbf{u}^i \leq V_k \cdot d_k^{\text{out}} & \forall p_k \in P^B \end{cases}$$
- 5 Determine all the next timed events, select the nearest in time and deduce time τ_{i+1} ;
- 6 Determine the new marking \mathbf{m}^{i+1} and the marking quantity vector \mathbf{q}^{i+1} ;
- 7 $i = i + 1$;
- 8 $\mathbf{u}^i = \boldsymbol{\varphi}^s$;
- 9 **Return** $(\mathbf{u}^0, \tau_0), \dots, (\mathbf{u}^i, \tau_i)$

Line 1 initializes the initial state as well as the control firing flow vector. The main part (lines 2 – 7) builds iteratively marking \mathbf{m}^i which is compared with the steady marking \mathbf{m}^s each timed event, and the control trajectory is computed by controlling the firing flow vector. Line 3 determines the necessary sets for the constraints to solve the linear programming problem (LPP). For the LPP part, the objective is to maximize the controlled firing flow vector at each timed event. Constraints (a)

show that the maximal flow of each transition t_j cannot exceed the steady firing flow φ_j^s . Constraints (b) block transition t_j when it is not enabled or when the marking quantity of one of its input places is lower than its steady marking quantity. Constraints (c) are used to decrease marking quantity to the steady marking quantity, while constraints (d) hold the marking quantity to the steady marking quantity. Constraints (e) require that the output flow of a batch place should be lower than or equal to the flow out coming from the place. Lines 5–6 consider the nearest event and update the new marking. The procedure is repeated until the steady marking is reached. When the marking \mathbf{m}^i reaches the steady value \mathbf{m}^s , the controlled firing flow vector is also set to its steady firing flow vector φ^s as shown in line 8. The complexity of this algorithm is polynomial since it is obtained by solving a LPP at each timed step.

Remark 2. For transitions that are in structural conflict, some constraints could be added for solving these conflicts when there are effective ones. For instance, with a proportional policy depending on the steady state value, the following constraints could be added as $u_a^i \cdot \varphi_b^s = u_b^i \cdot \varphi_a^s$ if $\forall t_a, t_b \in T, \bullet t_a \cap \bullet t_b \neq \emptyset$ and $t_a, t_b \notin T_{\mathcal{N}}(\mathbf{m}^i) \cup T_L(\mathbf{m}^i)$.

4.2 Transient behavior of places in a controlled net

In this section, we address the problem of the convergence of the marking to its steady state when the net is controlled by the proposed strategy. First, we consider that at a given time, the marking quantity vector is different from the steady marking quantity vector. In this case, we show that each place converges to its steady quantity. Next, it is proved that when the steady quantity is reached, the steady marking could also be reached in finite time. Let us first introduce some notations and results that apply for batch places.

We denote by δ_i^{\min} the minimum delay required to cross batch place p_i at speed V_i , defined by: $\delta_i^{\min} = s_i/V_i$.

At the steady state, the input flow and the output flow of place p_i are equal, i.e., $\phi_i^s = \text{Post}(p_i, \cdot) \cdot \varphi^s = \text{Pre}(p_i, \cdot) \cdot \varphi^s$.

According to the proposed control strategy, a first result on the density of created batches can be established.

Lemma 9. Consider a cGBPN $\langle N, \mathbf{m}_0 \rangle$ that satisfies assumptions (A1) – (A3) and is controlled by Algorithm 1 for reaching its steady state $(\mathbf{m}^s, \varphi^s)$. From the initial time, any created batch $\beta_k(\tau) = (0, d_k(\tau), 0)$ of batch place p_i , has its density constrained by: $d_k(\tau) \leq d_i^s$ with $d_i^s = \phi_i^s/V_i$.

Proof. At a time $\tau \geq \tau_0$, when a batch is created in batch place p_i , i.e., $\beta_k(\tau) = (0, d_k(\tau), 0)$, its density is given by: $d_k(\tau) = \phi_i^{\text{in}}(\tau)/V_i = \text{Post}(p_i, \cdot) \cdot \mathbf{u}(\tau)/V_i$. As the proposed control strategy imposes the firing flow to be lower than or equal to its steady state value, i.e., $\mathbf{u}(\tau) \leq \varphi^s$, it holds: $d_k(\tau) \leq \phi_i^s/V_i$. \diamond

Let us now focus on the transient behavior of places in a controlled net.

Lemma 10. Consider a cGBPN $\langle N, \mathbf{m}_0 \rangle$ that satisfies assumptions (A1) – (A3) and is controlled by Algorithm 1 for reaching its steady state $(\mathbf{m}^s, \varphi^s)$. Every place p_i of marking quantity not equal to its steady marking quantity

at time τ , i.e., $q_i(\tau) \neq q_i^s$, converges to its steady marking quantity q_i^s in a finite time.

Proof. According to assumption (A3), every place p_i belongs to the support of a positive P-semiflow, i.e., $\exists \mathbf{X} > 0$ such that $\mathbf{X}^T \cdot \mathbf{C} = \mathbf{0}$ and $p_i \in \|\mathbf{X}\|, \forall p_i \in P$. Thus, at time τ , the marking quantity verifies: $x_1 \cdot q_1(\tau) + \dots + x_n \cdot q_n(\tau) = x_1 \cdot q_1^0 + \dots + x_n \cdot q_n^0 = x_1 \cdot q_1^s + \dots + x_n \cdot q_n^s$.

a) For places with marking quantities strictly lower than their steady marking quantities, i.e., $q_i(\tau) < q_i^s$, the control strategy enforces OFF to their output transitions, i.e., $u_j(\tau) = 0$ (constraints (b)). Consequently, from time τ , their output flows remain null, $\phi_i^{\text{out}} = 0$, and their marking quantities increase in time (i.e., $\dot{q}_i = \phi_i^{\text{in}}$) until their steady quantities.

b) For places with marking quantities strictly greater than their steady marking quantities, i.e., $q_i(\tau) > q_i^s$, the control strategy enforces their input flows to be lower than or equal to their output flows (constraints (c)). This implies that their quantities could not increase, as $\dot{q}_i = \phi_i^{\text{in}} - \phi_i^{\text{out}}$ with $\phi_i^{\text{in}} \leq \phi_i^{\text{out}}$. As all places belong to the same P-invariant, it must exist some other places with a lower quantity than its steady quantity. As previously seen in case a, the quantity of these latter places increase in time. Consequently, to be conservative, the marking quantities of the former places decrease in time until their steady quantities.

c) For places with marking quantities equal to their steady marking quantities, i.e., $q_i(\tau) = q_i^s$, the control strategy enforces their input flows and output flows to be equal (constraints (d)). Thus, $\dot{q}_i = 0$. Such places have still reached their steady quantity. \diamond

Lemma 11. Consider a cGBPN $\langle N, \mathbf{m}_0 \rangle$ that satisfies assumptions (A1) – (A3) and is controlled by Algorithm 1 for reaching its steady state $(\mathbf{m}^s, \varphi^s)$. If at time τ , the marking quantity vector equals to its steady marking quantity vector, i.e., $\mathbf{q}(\tau) = \mathbf{q}^s$, the marking reaches the steady state marking \mathbf{m}^s in a finite time.

Proof. At time τ , all marking quantities equal to their steady marking quantities, i.e., $q_i(\tau) = q_i^s$. Due to constraints (d), the steady state quantities of places will remain constant from date τ . For continuous places, their steady state markings already correspond to their quantities. Thus, let us focus on the behavior of batch places.

For batch place p_i , at the steady state it holds: $\phi_i^{\text{in}} = \phi_i^{\text{out}} = \phi_i^s$ and its steady state marking m_i^s has one of the regular forms given by Proposition 6. According to assumption (A2), it holds $\phi_i^s > 0$ and consequently, the steady marking could not be in regular form (1). If $q_i(\tau) = q_i^s = Q_i$, the regular form $m_i^s = \{(s_i, d_i^{\max}, s_i)\}$ is already reached for place p_i . Thus, let us focus on the two other forms: i) $m_i^s = \{(s_i, d_i^s, s_i)\}$ and ii) $m_i^s = \{(l_e, d_i^s, l_e), (s_i - l_e, d_i^{\max}, s_i)\}$ with $d_i^s = \phi_i^s/V_i$.

Let us consider at date τ , the general case where the marking of batch place p_i is given by: $m_i(\tau) = \{\beta_1(\tau), \dots, \beta_\gamma(\tau)\}$ ordered by their head positions with $\sum_{k=1}^{\gamma} l_k(\tau) \cdot d_k(\tau) = q_i^s$.

From Lemma 9, any new created batch from date τ in p_i has a density lower than or equal to d_i^s . At date $\tau' = \tau + \max_{p_i \in P^B} \{\delta_i^{\min}\}$, all batches of the net have had time to

traverse their batch place. Thus, if at time τ' , place p_i has not reached its steady marking, it has a new marking $m_i(\tau') = \{\beta'_1(\tau'), \dots, \beta'_{\gamma'}(\tau')\}$ such that $d_k(\tau') \leq d_i^s$ for $k = 1, \dots, (\gamma' - 1)$ and necessarily $\beta'_{\gamma'}(\tau')$ is a dense output batch, i.e., $d_{\gamma'}(\tau') = d_i^{\max}$.

From date τ' , all the input and output flows of the places could be set to their steady state flow since the output density of batch places is greater than or equal to the steady state value. Moreover, we show that this steady flow is maintained until the steady state marking is reached for all the batch places. To do so, we show that the output density d_i^{out} of all batch places will remain greater than or equal to the steady state density d_i^s whatever the regular form that must be reached.

a) The steady marking of p_i is of form (i) $m_i^s = \{(s_i, d_i^s, s_i)\}$. From date τ' where the steady flow vector is set, a new batch $\beta'_0(\tau')$ of density d_i^s is created. From this date, all batches from $\{\beta'_1(\tau'), \dots, \beta'_{\gamma'-1}(\tau')\}$ move inside the place, meet the output batch and are merged with the dense output batch. As the quantity remains constant (i.e., $\dot{q}_i = 0$), the dense output batch cannot be destroyed until the length of the new created batch β'_0 becomes s_i in a delay that does not exceed δ_i^{\min} and thus the steady marking m_i^s is reached. As a consequence, the output density satisfies $d_i^{\text{out}} \geq d_i^s$ from date τ' .

b) The steady marking of p_i is of form (ii) $m_i^s = \{(l_e, d_i^s, l_e), (s_i - l_e, d_i^{\max}, s_i)\}$. At date τ' , the conservation law on quantities implies: $\sum_{k=1}^{\gamma'} l'_k(\tau') \cdot d'_k(\tau') = l_e \cdot d_i^s + (s_i - l_e) \cdot d_i^{\max} = q_i^s$. Thus, it acts that: $\sum_{k=1}^{\gamma'-1} l'_k(\tau') \cdot d'_k(\tau') < l_e \cdot d_i^s$ and $l_{\gamma'}(\tau') \cdot d_i^{\max} > (s_i - l_e) \cdot d_i^{\max}$. Consequently, the length of its dense output batch verifies: $l_{\gamma'}(\tau') \geq s_i - l_e$. From date τ' , the steady state flow vector is set and a new batch with density d_i^s is created. As in the previous case, the dense output batch cannot be destroyed and the new created one moves freely while intermediate batches accumulate with the output batch. This latter ensures an output density $d_i^{\text{out}} = d_i^{\max}$ until reaching the regular form m_i^s when the created batch meets the output batch in a delay that does not exceed δ_i^{\min} .

According to the considered cases, all batch places have an output density that satisfies $d_i^{\text{out}} \geq d_i^s$ from date τ' . We get to the fact that the vector ϕ_i^s is maintained and the marking converges to the steady state marking \mathbf{m}^s in a finite delay at time $\tau_s \leq \tau + 2 \cdot \max_{p_i \in PB} \{\delta_i^{\min}\}$. \diamond

Remark 3. Another way to reach the steady state \mathbf{m}^s from a state where $\mathbf{q}(\tau) = \mathbf{q}^s$, is to proceed within two steps (Demongodin and Giua, 2014): in the first one, from $\mathbf{m}(\tau)$ blocking all transitions by setting the controlled flow vector to zero until each quantities accumulates in a single output dense batch. In a second step, setting the controlled flow vector to the steady state one, $\mathbf{u} = \boldsymbol{\varphi}^s$ and wait until the marking \mathbf{m}^s is reached. This strategy imposes a null control vector during the accumulation mode where the already obtained regular forms are lost and should be reconstructed in the second step introducing more delay, hence this approach is not used in the present work.

Based on the above results, the convergence of the proposed algorithm could be analyzed.

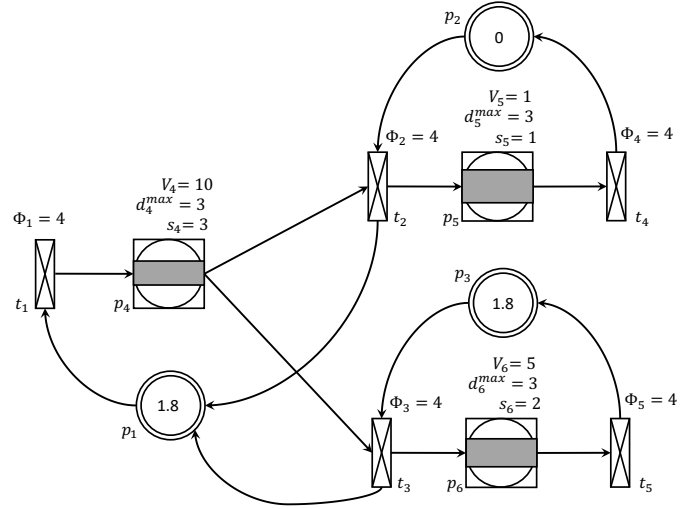


Fig. 3. The cGBPN of Example 7 in a steady state.

Proposition 12. Given a cGBPN $\langle N, \mathbf{m}_0 \rangle$ that satisfies assumptions (A1) – (A3). Algorithm 1 computes a control trajectory that drives the system from the initial state to a given reachable steady state $(\mathbf{m}^s, \boldsymbol{\varphi}^s)$ in finite time.

Proof. According to Lemma 10, in a finite time, the cardinality of sets $S_G(\mathbf{m}^i)$ and $S_L(\mathbf{m}^i)$ decreases while the one of set $S_E(\mathbf{m}^i)$ increases until $S_E(\mathbf{m}^i) = P$. According to Lemma 11, Algorithm 1 converges to the steady state in finite time. \diamond

4.3 Example

Let us consider again the cGBPN of Fig. 2 with $\mathbf{m}_0 = [3 \ 0.4 \ 3 \ \emptyset \ \{(0.2, 3, 1)\} \ \emptyset]^T$. As previously stated, a reachable steady state $(\mathbf{m}^s, \boldsymbol{\varphi}^s)$ is given by $\mathbf{m}^s = [1.8 \ 0 \ 1.8 \ \{(3, 0.4, 3)\} \ \{(1, 1, 1)\} \ \{(2, 0.6, 2)\}]^T$ as shown in Fig. 3 and, $\boldsymbol{\varphi}^s = [4 \ 1 \ 3 \ 1 \ 3]^T$. The steady marking quantity vector is equal to $\mathbf{q}^s = [1.8 \ 0 \ 1.8 \ 1.2 \ 1 \ 1.2]^T$.

Under the proposed strategy, a control trajectory that drives the net from its initial state to the steady state is obtained as follows.

- At $\tau_0 = 0$, $T_N(\mathbf{m}_0) = \{t_2, t_3, t_5\}$. As $\mathbf{q}^0 = [3 \ 0.4 \ 3 \ 0 \ 0.6 \ 0]^T$, the subsets of places are: $S_E(\mathbf{m}_0) = \emptyset$, $S_G(\mathbf{m}_0) = \{p_1, p_2, p_3\}$ and $S_L(\mathbf{m}_0) = \{p_4, p_5, p_6\}$. Thus, $T_L(\mathbf{m}_0) = \{t_2, t_3, t_4, t_5\}$. Consequently, only transition t_1 is enabled and ON. By computing the LPP in Algorithm 1, the controlled firing flow vector is $\mathbf{u}^0 = [4 \ 0 \ 0 \ 0 \ 0]^T$.
- At $\tau_1 = 0.3$, places p_1 and p_4 reach their steady marking quantities. At this date, the marking is $\mathbf{m}^1 = [1.8 \ 0.4 \ 3 \ \{(3, 0.4, 3)\} \ \{(0.2, 3, 1)\} \ \emptyset]^T$ and $\mathbf{q}(\tau_1) = [1.8 \ 0.4 \ 3 \ 1.2 \ 0.6 \ 0]^T$. This implies: $T_N(\mathbf{m}^1) = \{t_5\}$, $S_E(\mathbf{m}^1) = \{p_1, p_4\}$, $S_G(\mathbf{m}^1) = \{p_2, p_3\}$, $S_L(\mathbf{m}^1) = \{p_5, p_6\}$ and $T_L(\mathbf{m}^1) = \{t_4, t_5\}$. Transitions t_1, t_2 and t_3 are enabled and ON since the marking quantities of their input places are greater than or equal to their steady marking quantities. Since p_4 is in $S_E(\mathbf{m}^1)$, constraints (d) imposes $u_1^1 = u_2^1 + u_3^1$. Finally, by maximizing all the controlled flows, the LPP gives: $\mathbf{u}^1 = [4 \ 1 \ 3 \ 0 \ 0]^T$.

- At $\tau_2 = 0.7$, all places reach their steady marking quantities since $\mathbf{q}(\tau_2) = [1.8 \ 0 \ 1.8 \ 1.2 \ 1 \ 1.2]^T$, i.e., $\mathbf{m}^2 = [1.8 \ 0 \ 1.8 \ \{(3, 0.4, 3)\} \ \{(0.4, 1, 0.4), (0.2, 3, 1)\} \ \{(2, 0.6, 2)\}]^T$. Thus, the subsets of places are $S_E(\mathbf{m}^2) = \{p_1, p_2, p_3, p_4, p_5, p_6\}$, $S_G(\mathbf{m}^2) = S_L(\mathbf{m}^2) = T_N(\mathbf{m}^2) = T_L(\mathbf{m}^2) = \emptyset$. The computed controlled flow vector is: $\mathbf{u}^2 = [4 \ 1 \ 3 \ 1 \ 3]^T$. At this date, we can remark that the steady marking quantities and the steady flows are reached, i.e., we have $(\mathbf{q}^s, \varphi^s)$. Nevertheless, the steady state marking \mathbf{m}^s is not reached and next events must be considered.
- At $\tau_3 = 1.3$, the output density of p_5 changes from 3 to 1 (i.e., previous output batch (0.2, 3, 1) is destroyed and batch (1, 1, 1) becomes an output batch). Since, at this date, the marking is $\mathbf{m}^3 = [1.8 \ 0 \ 1.8 \ \{(3, 0.4, 3)\} \ \{(1, 1, 1)\} \ \{(2, 0.6, 2)\}]^T = \mathbf{m}^s$, the steady state is reached and the algorithm ends.

Finally, the resulting control trajectory is given by $(\mathbf{u}^0, 0)$, $(\mathbf{u}^1, 0.3)$, $(\mathbf{u}^2, 0.7)$, $(\mathbf{u}^3, 1.3)$ with,

$$\mathbf{u}^0 = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{u}^1 = \begin{bmatrix} 4 \\ 1 \\ 3 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{u}^2 = \begin{bmatrix} 4 \\ 1 \\ 3 \\ 1 \\ 3 \end{bmatrix} \quad \text{and} \quad \mathbf{u}^3 = \begin{bmatrix} 4 \\ 1 \\ 3 \\ 1 \\ 3 \end{bmatrix}.$$

Let us recall that the sum of firing quantities of transitions during time interval $[\tau_0, \tau_3]$ is given by:

$$\mathbf{z} = \int_{\tau_0}^{\tau_1} \mathbf{u}^0 \cdot d\rho + \int_{\tau_1}^{\tau_2} \mathbf{u}^1 \cdot d\rho + \int_{\tau_2}^{\tau_3} \mathbf{u}^2 \cdot d\rho.$$

From the obtained control trajectory, it holds:

$$\mathbf{z} = 0.3 \cdot \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0.4 \cdot \begin{bmatrix} 4 \\ 1 \\ 3 \\ 0 \\ 0 \end{bmatrix} + 0.6 \cdot \begin{bmatrix} 4 \\ 1 \\ 3 \\ 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 5.2 \\ 1 \\ 3 \\ 0.6 \\ 1.8 \end{bmatrix}.$$

From the fundamental equation Eq.(1), the marking quantity at the date of the steady state is:

$$\mathbf{q}(1.3) = \begin{bmatrix} 3 \\ 0.4 \\ 3 \\ 0 \\ 0.6 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 5.2 \\ 1 \\ 3 \\ 0.6 \\ 1.8 \end{bmatrix} = \begin{bmatrix} 1.8 \\ 0 \\ 1.8 \\ 1.2 \\ 1 \\ 1.2 \end{bmatrix}.$$

Let us remark that $C(\mathbf{z} - \mathbf{z}^s) = 0$ where $\mathbf{z}^s = [2.8 \ 0.4 \ 1.2 \ 0 \ 0]^T$.

The firing quantity \mathbf{z}^s is the minimal quantity that must be fired to reach the steady state quantities and $(\mathbf{z} - \mathbf{z}^s)$ is the quantity fired to reach the steady state marking \mathbf{m}^s from the steady marking quantities.

5. CONCLUSIONS

An event-driven control strategy for generalized batches Petri nets has been studied in this work, which could be

applied to analyze the high throughput production lines or transportation networks. An algorithm that computes a control trajectory for reaching a given steady state from an initial one has been proposed. By analyzing the transient behavior of places in a controlled net, the convergence of the algorithm in finite time to the target steady state has been proved. Based on the control strategy, the controlled firing flow cannot exceed the steady firing flow. Due to this limitation, the time optimality of the convergence cannot be ensured. Moreover, only steady states are considered as target markings to be reached. Our future works would extend the presented one to the design of a minimum-time control strategy. We will also consider the problem of reaching any given reachable marking from an initial one and will address the control for cGBPNs with discrete nodes.

REFERENCES

- Aguayo-Lara, E., Ross-León, R., Ramírez-Treviño, A., and Ruiz-León, J. (2012). On/Off control with observer's state feedback for continuous timed Petri nets. *IFAC Proceedings Volumes*, 45(29), 221–226.
- Cassandras, C.G. (2014). The event-driven paradigm for control, communication and optimization. *Journal of Control and Decision*, 1(1), 3–17.
- David, R. and Alla, H. (2005). *Discrete, continuous, and hybrid Petri nets*, volume 1. Springer.
- Demongodin, I. and Giua, A. (2014). Dynamics and steady state analysis of controlled generalized batches Petri nets. *Nonlinear Analysis: Hybrid Systems*, 12, 33–49.
- Demongodin, I. (2001). Generalised batches Petri net: hybrid model for high speed systems with variable delays. *Discrete Event Dynamic Systems*, 11(1-2), 137–162.
- Demongodin, I. and Giua, A. (2010). Linear programming techniques for analysis and control of batches Petri nets. In *10th International Workshop on Discrete Event Systems, WODES 2010, Berlin, Germany, August 30 - September 01*, 1–6.
- Demongodin, I. and Giua, A. (2012). Stationary behavior of controlled generalized batches Petri nets. *IFAC Proceedings Volumes*, 45(29), 54–60.
- Júlvez, J., Di Cairano, S., Bemporad, A., and Mahulea, C. (2014). Event-driven model predictive control of timed hybrid Petri nets. *International journal of robust and nonlinear control*, 24(12), 1724–1742.
- Júlvez, J., Recalde, L., and Silva, M. (2003). On reachability in autonomous continuous Petri net systems. In *International Conference on Application and Theory of Petri Nets*, 221–240. Springer.
- Mahulea, C., Giua, A., Recalde, L., Seatzu, C., and Silva, M. (2008). Optimal model predictive control of timed continuous Petri nets. *IEEE Transactions on Automatic Control*, 53(7), 1731–1735.
- Navarro-Gutiérrez, M., Ramírez-Treviño, A., and Silva, M. (2017). Homothecy, bifurcations, continuity and monotonicity in timed continuous Petri nets under infinite server semantics. *Nonlinear Analysis: Hybrid Systems*, 26, 48–67.
- Wang, L., Mahulea, C., Júlvez, J., and Silva, M. (2014). ON/OFF strategy based minimum-time control of continuous Petri nets. *Nonlinear Analysis: Hybrid Systems*, 12, 50–65.