A Control Method for Timed Distributed Continuous Petri nets

Hanife Apaydın-Özkan, Jorge Júlvez, Cristian Mahulea and Manuel Silva

***Abstract*— A timed distributed continuous Petri net system (DcontPN) is composed of several subsystems which communicate through channels modeled by places. In this work, a reachability control problem for DcontPNs composed of two subsystems is considered. An algorithm is developed to calculate the control inputs for each subsystem. The application of the obtained control inputs drives the subsystems from the initial states to the target states in a finite amount of time. The algorithm allows the subsystems to reach their respective target markings at different time instants and keep them as long as required.**

I. INTRODUCTION

A recurrent problem in large discrete event systems is the state explosion problem. This well known problem makes the use of many analysis and verification techniques computationally prohibitive when applied to many systems of interest in practice. One way of avoiding such a problem is to relax the original discrete model and deal with a continuous approximation of it. In the Petri nets framework this leads to continuous Petri nets [3], [4], [2].

This work proposes a control strategy for timed distributed continuous Petri nets (DContPN). A DContPN is composed of several subsystems interconnected by means of communication channels. In order to reach a target state (or marking), subsystems may require data from the other subsystems. These data is sent by means of the communication channels. The underlying idea of the strategy is to design a local controller for each subsystem. Each controller computes the control actions required to reach a given target state independently, and asks the other controllers to produce enough data in the communication channels to execute its control actions. This paper mainly focuses on reaching the target markings of all subsystems in a finite amount of time. This does not imply that the time instants to reach the target marking in each subsystem are the same but only that after some time all subsystems reach their desired marking and stay there.

Some works have been done in literature about distributed systems. For example, distributed timed automata is defined in [5] and a supervisory control system for a distributed manufacturing process is studied by using discrete Petri nets [1]. An architecture for distributed implementation of Petri nets in control applications is proposed in [10]. But, as far as authors knowledge this study is the first work about the control of DcontPN.

The remainder of the paper is organized as follows: Section II introduces timed distributed continuous Petri nets. The control problem under consideration is presented in Section III. Section IV proposes a control algorithm for a particular class of DcontPN. A case study consisting of a distributed manufacturing system is shown in Section V. Section VI summarizes the main conclusions of the work.

II. DISTRIBUTED CONTINUOUS PETRI NETS

This section introduces the main concepts related to DcontPN and presents an introductory example. The reader is assumed to be familiar with basic Petri net concepts (see [8] for a gentle introduction).

***Definition** 2.1 (contPN):* A continuous Petri net (contPN) $\mathcal{N}$ is a tuple $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}, \boldsymbol{\lambda} \rangle$ where:

- P and T are the sets of places and transitions respectively.
- $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{R}_{\geq 0}^{|P|\times|T|}$ are the pre and post incidence matrices.
- $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{|T|}$ is the firing rate of transition.

Let p_i, $i = 1, \ldots, |P|$ and $t_j, j = 1, \ldots, |T|$ denote the places and transitions. For a place $p_i \in P$ and a transition $t_j \in T$, $Pre_{ij} = Pre(p_i, t_j)$ and $Post_{ij} = Post(p_i, t_j)$ represent the weights of the arcs from p_i to t_j and from t_j to p_i, respectively. Each place p_i has a marking denoted by $m_i \in \mathbb{R}_{\geq 0}$. The vector of all token loads is called *state* or *marking*, and is denoted by $\boldsymbol{m} \in \mathbb{R}_{\geq 0}^{|P|}$. For every node $v \in P \cup T$, the sets of its input and output nodes are denoted as ${}^\bullet v$ and $v^\bullet$, respectively.

A transition $t_j \in T$ is enabled at $\boldsymbol{m}$ iff $\forall p_i \in^\bullet t_j$, $m_i > 0$ and its enabling degree is given by

$$enab(t_j, \boldsymbol{m}) = \min_{p_i \in {}^\bullet t_j} \left\{ \frac{m_i}{Pre_{ij}} \right\}$$

which represents the maximum amount in which t_j can fire. 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 $\boldsymbol{m'} = \boldsymbol{m} + \alpha \cdot \boldsymbol{C}_{\cdot j}$ where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the token flow matrix and $\boldsymbol{C}_{\cdot j}$ is its j^{th} column. 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 $\boldsymbol{\sigma} \in \mathbb{R}_{\geq 0}^{|T|}$ is the firing count vector, i.e., σ_j is the cumulative amount of firings of t_j in the sequence σ.

This work was partially supported by CICYT - FEDER projects DPI2006-15390, TIN2007-66523, and by the European Communitys Seventh Framework Programme under project DISC (Grant Agreement n. INFSO-ICT-224498).

The authors are with Instituto de Investigación en Ingeniería de Aragon (I3A), Universidad de Zaragoza, Spain hapaydin,julvez,cmahulea,silva@unizar.es

The work of H. Apaydin-Özkan was supported in part by the Diputación General de Aragón for one year stay in the I3A; on leave from Faculty of Electrical and Electronics Engineering, Anadolu University, Eskisehir, Turkey.

***Definition** 2.2 (contPN system):* A contPN system is a pair $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ where $\mathcal{N}$ is a contPN and $\boldsymbol{m}_0 \in \mathbb{R}_{\geq 0}^{|P|}$ is the initial marking.

The state equation has an explicit dependence on time, denoted by τ: $\boldsymbol{m}(\tau) = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}(\tau)$ which through time differentiation becomes $\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \dot{\boldsymbol{\sigma}}(\tau)$. The derivative of the firing sequence $\boldsymbol{f}(\tau) = \dot{\boldsymbol{\sigma}}(\tau)$ is called the firing flow. Depending on how the flow is defined, many firing semantics appear, being the most used ones *infinite* and *finite* server semantics. For a broad class of Petri nets it is shown that infinite server semantics offers better approximation than finite server semantics [7]. This paper deals with infinite server semantics for which the flow of a transition t_j is defined as:

$$f_j(\tau) = \lambda_j \cdot enab(t_j, \boldsymbol{m}(\tau)) = \lambda_j \cdot \min_{p_i \in {}^\bullet t_j} \left\{ \frac{m_i(\tau)}{Pre_{ij}} \right\} \quad (1)$$

Left and right natural annullers of the token flow matrix $\boldsymbol{C}$ are called *P-semiflows* (denoted by $\boldsymbol{r}$) and *T-semiflows* (denoted by $\boldsymbol{s}$), respectively. If $\exists\ \boldsymbol{r} > 0,\ \boldsymbol{r} \cdot \boldsymbol{C} = 0$, then the net is said to be *conservative*. If $\exists\ \boldsymbol{s} > 0,\ \boldsymbol{C} \cdot \boldsymbol{s} = 0$ it is said to be *consistent*. The support of a vector $\boldsymbol{v}$ is the set of nonzero components and denoted by $\|\boldsymbol{v}\|$. A semiflow $\boldsymbol{v}$ is said to be *minimal* when its support, $\|\boldsymbol{v}\|$, is not a proper superset of any other, and the greatest common of its elements is one.

***Definition** 2.3 (MTS):* [7] A PN is mono T-semiflow (MTS) if it is conservative, consistent and has only one minimal T-semiflow.

***Definition** 2.4 (DcontPN):* A Distributed timed contPN (DcontPN) system is a set of contPN systems connected through channels modeled as places.

Let K denote the set of subsystems of a given DcontPN. The set of places and transitions of subsystem $k \in K$ is denoted by P^k and T^k, respectively. The token flow or incidence matrix of subsystem $k \in K$ is denoted by $\boldsymbol{C}^k \in \mathbb{R}^{|P^k| \times |T^k|}$.

We assume, $P^k \cap P^l = \emptyset$ and $T^k \cap T^l = \emptyset$, $\forall k, l \in K,\ k \neq l$. The directional communication between subsystems is provided by a set of places called *channel places*. In particular, the communication from subsystem k to l is provided by a set of places denoted $P^{k,l}$, whose input transitions are contained in subsystem k and output transitions are contained in subsystem l, i.e., $P^{k,l} = \{p \in P | {}^\bullet p \in T^k,\ p^\bullet \in T^l,\ p \notin P^q \ \ \forall q \in K\}$.

Note that $p \in P^{k,l}$ is an *input channel* of subsystem l and an *output channel* of subsystem k. The set of all output channels of subsystem k is denoted by $P^{k,*}$, i.e., $P^{k,*} = \bigcup_{l \in K, l \neq k} P^{k,l}$, and the set of all input channels of subsystem k is denoted by $P^{*,k}$, i.e., $P^{*,k} = \bigcup_{l \in K, l \neq k} P^{l,k}$. The marking vector of a subsystem k is denoted by $\boldsymbol{m}(P^k) \in \mathbb{R}_{\geq 0}^{|P^k|},\ \ \forall k \in K$. When designing a controller, it must be taken into account that the controller of a given subsystem can only know the marking of the input places of that subsystem, i.e., the marking of the input places of the other subsystems are not observable.

***Example** 1:* Let us consider the simple DcontPN given in Figure 1. It is composed of two subsystems. For the first subsystem, the sets of places and transitions are $P^1 = \{p_1,\ p_2,\ p_3,\ p_4,\ p_5\}$, $T^1 = \{t_1,\ t_2,\ t_3,\ t_4\}$; while they are $P^2 = \{p_6,\ p_7,\ p_8\}$ and $T^2 = \{t_5,\ t_6,\ t_7\}$ for the second one. These two subsystems communicate through two channels: p_a for the communication from subsystem 2 to subsystem 1 and p_b for the communication from subsystem 1 to subsystem 2. Hence, $P^{1,2} = \{p_b\}$ and $P^{2,1} = \{p_a\}$ implying $P^{*,1} = P^{2,*} = \{p_a\}$ and $P^{*,2} = P^{1,*} = \{p_b\}$. Finally, the token flow matrices of subsystems are,

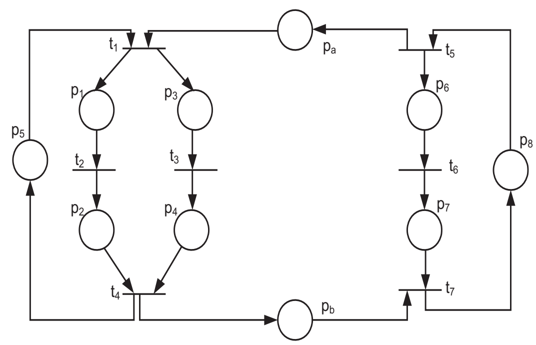

Fig. 1. A simple DcontPN

$$\boldsymbol{C}^1 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{C}^2 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}$$

III. Control of DcontPN

This section shows how control actions can be introduced to DcontPN, and establishes the control problem that is considered in the following sections.

A. Control Actions

The autonomous (or uncontrolled) behavior of a DcontPN described in the previous section can be modified by introducing control actions. In continuous Petri nets the control actions are applied on the transitions and they can only *slow-down* (never speed-up) the firing flow of the transitions to which they are applied [11].

***Definition** 3.1:* The controlled flow, $\boldsymbol{w}$, of a timed DcontPN is defined as $\boldsymbol{w}(\tau) = \boldsymbol{f}(\tau) - \boldsymbol{u}(\tau)$, with $0 \leq \boldsymbol{u}(\tau) \leq \boldsymbol{f}(\tau)$, where $\boldsymbol{f}$ is the flow of the uncontrolled system, i.e., defined as in (1), and $\boldsymbol{u}$ is the control action.

Therefore, the control input $\boldsymbol{u}$ is dynamically upper bounded by the flow $\boldsymbol{f}$ of the corresponding unforced system. Under these conditions, the overall behaviour of the system in which all transitions are controllable is ruled by the following system:

$$\begin{array}{l} \dot{\boldsymbol{m}} = \boldsymbol{C} \cdot [\boldsymbol{f} \ - \ \boldsymbol{u}] = \boldsymbol{C} \cdot \boldsymbol{w} \\ 0 \leq \boldsymbol{u} \leq \boldsymbol{f} \end{array} \quad (2)$$

The integral of the controlled flow of a transition t_j over an interval of time $(\tau_a,\ \tau_b)$ is denoted by $x(t_j) = \int_{t_a}^{t_b} w(t_j) d\tau$.

For the sake of clarity, τ will be omitted in the rest of the paper: $f(t_j)$ and $x(t_j)$ will be used instead of $f(t_j, \tau)$ and $x(t_j, \tau)$, respectively.

B. Problem Statement

Among the different existing control problems, we will deal with a reachability control problem. We will assume that each subsystem of a given DcontPN aims at reaching a particular target marking. In contrast with a centralized control, each subsystem is equipped with its own controller that computes the control actions that drive the subsystem to the target marking. Given that the subsystems are interconnected, subsystems may require data to be available in the communication channels to reach the target marking. The following example shows this situation.

***Example** 2:* Consider the DcontPN in Figure 1 with $\boldsymbol{m}_0(P^1) = [1\ 2\ 1\ 1\ 2]^T$, $\boldsymbol{m}_0(P^2) = [1\ 3\ 2]^T$, $m_0(p_a) = 0$, $m_0(p_b) = 1$ and $\boldsymbol{m}_f(P^1) = [2\ 2\ 2\ 1\ 1]^T$, $\boldsymbol{m}_f(P^2) = [1\ 3\ 2]^T$. A controller for the first subsystem could compute $x(t_1) = 1$, $x(t_2) = x(t_3) = x(t_4) = 0$ so that the subsystem reaches the target marking. Given that the initial marking and target marking of subsystem 2 are the same, the controller for that subsystem could yield $x(t_5) = x(t_6) = x(t_7) = 0$. Given that $m_0(p_a) = 0$, transition t_1 cannot fire unless t_5 fires. Unfortunately, according to the computed controls t_5 will not fire, since $x(t_5) = 0$. Hence, the computed controls are not valid to reach the desired target marking of subsystem 1. In order to solve this situation, subsystem 1 may ask subsystem 2 to put enough tokens in p_a. This can be achieved easily by firing t_5, however this will imply that subsystem 2 moves away from its desired target marking.

When stating the problem we are implicitly assuming that the target markings of each subsystem are required to be reached simultaneously. Other focuses, as reaching first one target marking and then the others, could be considered, however this is beyond the scope of this work. Apart from the problem of tokens required in the communication channels, it could happen that the target markings are not reachable simultaneously due to the system structure and target marking.

***Example** 3:* Consider again DcontPN in Figure 1. For subsystem 1, let the target marking be $\boldsymbol{m}_f(P^1) = [2\ 2\ 2\ 1\ 1]^T$ which is reachable from $\boldsymbol{m}_0(P^1) = [1\ 2\ 1\ 1\ 2]^T$ locally by firing t_1, i.e., if it is considered isolated from the rest of the system. For subsystem 2, let the target marking be $\boldsymbol{m}_f(P^2) = [1\ 3\ 2]^T$ which is reachable from $\boldsymbol{m}_0(P^2) = [1\ 3\ 2]^T$ locally. But when both subsystems are connected through the communication places p_a and p_b with $m_0(p_a) = 0$, $m_0(p_b) = 0$, the target markings are not reachable simultaneously.

Next subsection proposes a distributed controller that produces enough tokens in the communication channels and that provides a necessary and sufficient condition for the simultaneous reachability of the target markings.

IV. A CONTROLLER FOR DCONTPN

This section is devoted to the design of a distributed controller for a particular class of DcontPN. We will first present the algorithm associated to the local controller of each subsystem and then state two properties related to its correctness.

A. Design of a distributed controller

The control algorithm that will be presented applies to those DcontPN systems satisfying the following assumptions:

(A1) The DcontPN is composed of two subsystems that are MTS. The minimal T-semiflows are denoted by $\boldsymbol{s}^1$ and $\boldsymbol{s}^2$.

(A2) The target marking $\boldsymbol{m}_f$ is positive and reachable at the overall system.

(A3) The following equalities are satisfied $\forall p_a \in P^{2,1}$ $\forall p_b \in P^{1,2}$

$$\sum_{t \in p_b^\bullet} Pre(p_b, t) \cdot s^2(t) = \sum_{t \in {}^\bullet p_a} Post(p_a, t) \cdot s^2(t)$$
$$\sum_{t \in {}^\bullet p_b} Post(p_b, t) \cdot s^1(t) = \sum_{t \in p_a^\bullet} Pre(p_a, t) \cdot s^1(t) \tag{3}$$

The first assumption reduces the class of DcontPN to those systems having two subsystems that are MTS. The second assumption is simply a necessary condition for simultaneous reachability of the target markings (it will be shown that this reachability condition can be deduced from the output of the control algorithm). The third assumption states that for any couple of input and output channels, the number of tokens produced by the execution of a T-semiflow in the output channel must be equal to the number of tokens consumed from the input channel. Although these assumptions may seem to be restrictive, they are satisfied by a number of systems of interest in practice as Marked Graphs, ordinary DcontPNs where each places has one input and one output transition.

The proposed algorithm represents the local controller that will be executed in each subsystem separately and has 6 basic steps. The algorithm that will be executed for subsystem 1 is given as Algorithm 1. For the algorithm of subsystem 2; $P^{2,1}, P^{1,2}, h^1$ are changed to $P^{1,2}, P^{2,1}, h^2$, respectively. In step 1 of the algorithm, subsystems compute the flow integrals required to reach the target markings without taking into account the marking of the communication places. Step 2 calculates the amounts of tokens that must be produced in the input channels in order to be able of firing the computed flow integrals. These amounts of tokens are sent from one subsystem to the other in steps 3 and 4 (we assume there is a communication channel between subsystem that allows these amounts to be sent).

In step 5, it is computed, how many tokens are put in each output channel by the present control. If this value is negative, more tokens are needed to be produced in the communication channels and flow integrals must be recomputed. In the algorithms, instead of checking negativity of the mentioned value, we check whether the value is less than a nonnegative constant γ. This way, communication channels can always be kept to a minimum number of γ

tokens. This recomputation is achieved at LPP (5) in step 6, where an extra constraint is added to LPP (4) of step 1, in order to ensure that enough tokens are produced in the communication channels.

Algorithm 1 Control of DcontPN

Input: $\boldsymbol{C}^1$, $\boldsymbol{m}_0(P^1)$, $\boldsymbol{m}_f(P^1)$, $\boldsymbol{Pre}(P^1,T^1)$, $\boldsymbol{Post}(P^1,\ T^1)$

1) Solve

$$\begin{array}{ll} \min & \mathbf{1}^T \cdot \bar{\boldsymbol{x}} \\ \text{s.t.} & \boldsymbol{m}_f(P^1) - \boldsymbol{m}_0(P^1) = \boldsymbol{C}^1 \cdot \bar{\boldsymbol{x}}, \\ & \bar{\boldsymbol{x}} \geq 0 \end{array} \tag{4}$$

2) For every $p \in P^{2,1}$ calculate

$$q_p^{REQ} = \left(\sum_{t \in p^\bullet} Pre(p,t) \cdot \bar{x}(t) \right) - m_0(p), \quad \forall p \in P^{2,1}$$

3) Send q_p^{REQ}, $\forall p \in P^{2,1}$ to the other subsystem

4) Receive r_p^{REQ}, $\ \forall p \in P^{1,2}$ from the other subsystem

5) Calculate

$$h_p^1 = \left(\sum_{t \in {}^\bullet p} Post(p,t) \cdot \bar{x}(t) \right) - r_p^{REQ}, \forall p \in P^{1,2}$$

6) **If** $\min\limits_{p \in P^{1,2}} \{h_p^1\} < \gamma$ **then** solve

$$\begin{array}{ll} \min & \mathbf{1}^T \cdot \boldsymbol{x} \\ \text{s.t.} & \boldsymbol{m}_f(P^1) - \boldsymbol{m}_0(P^1) = \boldsymbol{C}^1 \cdot \boldsymbol{x}, \\ & \left(\sum_{t \in {}^\bullet p} Post(p,t) \cdot x(t) \right) \geq r_p^{REQ} + \gamma, \ \forall p \in P^{1,2} \\ & \boldsymbol{x} \geq 0 \end{array} \tag{5}$$

Else

$$\boldsymbol{x} = \bar{\boldsymbol{x}}$$

End

Example 4: Let us go back to the DcontPN in Figure 1 to illustrate the process of the algorithms. Assume: $\boldsymbol{m}_0(P^1) = [1\ 2\ 1\ 1\ 2]^T$, $\boldsymbol{m}_0(P^2) = [1\ 3\ 2]^T$, $m_0(p_a) = 0,\ m_0(p_b) = 1$, $\boldsymbol{m}_f(P^1) = [2\ 2\ 2\ 1\ 1]^T$, $\boldsymbol{m}_f(P^2) = [1\ 3\ 2]^T$, and the nonnegative constant γ is set to $\gamma = 0$. The steps of the algorithm are shown in Table I.

Once the flow integral vectors $\boldsymbol{x}$ of the evolution from the initial marking to the target marking have been computed, the value of the control actions $\boldsymbol{u}$ can be derived in several ways as long as $\boldsymbol{x} = \int_{t_a}^{t_b} (\boldsymbol{f} - \boldsymbol{u}) d\tau$ is satisfied. Remark that $\boldsymbol{x}$ can be seen as a firing count vector in the untimed system and the problem of finding a control law $\boldsymbol{u}$ is equivalent to a reachability problem: if the desired marking is reachable in the untimed net system it is reachable in the timed one with an appropriate control law if all transitions are controllable. This result is proved in [6] (Prop. 14. 3) where a procedure that executes a firing sequence of the untimed system in the timed one is also presented.

B. Reachability of the target marking

This subsection presents two important results related to the presented algorithm. The first one (see Theorem 4.1) states that if the DcontPN satisfies the three assumptions (A1), (A2) and (A3) the algorithm is correct and will yield a control law that drives the system to the desired target marking. The second one (see Theorem 4.2) establishes a necessary and sufficient condition for the reachability of the target marking.

Step	Subsys. 1	Subsys. 2
Step1	$\bar{x}(t_1) = 1, \bar{x}(t_2) = 0$ $\bar{x}(t_3) = 0, \bar{x}(t_4) = 0$	$\bar{x}(t_5) = 0, \bar{x}(t_6) = 0$ $\bar{x}(t_7) = 0$
Step2	$q_{p_a}^{REQ} = \bar{x}(t_1) - m_0(p_a)$ $= 1$	$q_{p_b}^{REQ} = \bar{x}(t_7) - m_0(p_b)$ $= -1$
Step3	Send $q_{p_a}^{REQ} = 1$	Send $q_{p_b}^{REQ} = -1$
Step4	Receive $r_{p_b}^{REQ} = -1$	Receive $r_{p_a}^{REQ} = 1$
Step5	$h_{p_b}^1 = \bar{x}(t_4) - r_{p_b}^{REQ}$ $= 1$	$h_{p_a}^2 = \bar{x}(t_5) - r_{p_a}^{REQ}$ $= -1$
Step6	Given that $h_{p_b}^1 \geq \gamma = 0$, $x(t_1) = \bar{x}(t_1) = 1$ $x(t_2) = \bar{x}(t_2) = 0$ $x(t_3) = \bar{x}(t_3) = 0$ $x(t_4) = \bar{x}(t_4) = 0$	Given that $h_{p_a}^2 < \gamma = 0$, LPP (5) has to be solved, then: $x(t_5) = 1$ $x(t_6) = 1$ $x(t_7) = 1$

TABLE I

EXECUTION OF THE ALGORITHMS ON THE DCONTPN IN FIGURE 1

Theorem 4.1: Let $\mathcal{N}$ be a DcontPN satisfying assumptions (A1), (A2) and (A3). Algorithm 1 computes a control law that:

- drives the subsystems from $\boldsymbol{m}_0(P^1)$ and $\boldsymbol{m}_0(P^2)$ to target markings $\boldsymbol{m}_f(P^1)$ and $\boldsymbol{m}_f(P^2)$, simultaneously.
- the final markings of channels satisfy $m_f(p) \geq \gamma\ \forall p \in P^{1,2} \cup P^{2,1}$

Proof: See Appendix ■

Theorem 4.2: Let $\mathcal{N}$ be a DcontPN satisfying assumptions (A1) and (A3). Algorithm 1 computes a control law that:

- drives the subsystems from $\boldsymbol{m}_0(P^1)$ and $\boldsymbol{m}_0(P^2)$ to target markings $\boldsymbol{m}_f(P^1)$ and $\boldsymbol{m}_f(P^2)$, simultaneously.
- the final markings of channels satisfy $m_f(p) \geq \gamma\ \forall p \in P^{1,2} \cup P^{2,1}$

iff the target marking is reachable.

Proof: See Appendix ■

Example 5: Consider again DcontPN in Figure 1. Let the initial and target marking be: $\boldsymbol{m}_0(P^1) = [1\ 2\ 1\ 1\ 2]^T$, $\boldsymbol{m}_0(P^2) = [1\ 3\ 2]^T$, $m_0(p_a) = 0$, $m_0(p_b) = 0$ and $\boldsymbol{m}_f(P^1) = [2\ 2\ 2\ 1\ 1]^T$, $\boldsymbol{m}_f(P^2) = [1\ 3\ 2]^T$ and assume $\gamma = 0$. After the execution of the Algorithm 1, the control law of subsystem 1 provides $x(t_1) = 1,\ x(t_2) = x(t_3) = x(t_4) = 0$ while that of subsystem 2 provides $x(t_5) = x(t_6) = x(t_7) = 1$. According to these flow integral values, the controller of subsystem 2 tries to fire transition t_7

($x(t_7) = 1$). But it is not implementable, because $m_0(p_b) = 0$ and the local controller of subsystem 1 does not put any token to $p_b =^{\bullet} t_7$. Since one of the controller can not implement the computed control law, it is concluded that $\boldsymbol{m}_f(P^1)$ and $\boldsymbol{m}_f(P^2)$ are not reachable simultaneously from $\boldsymbol{m}_0(P^1)$ and $\boldsymbol{m}_0(P^2)$ while $m_0(p_a) = m_0(p_b) = 0$ with the request of at least $\gamma = 0$ number of tokens at the channels.

But if we have assumed $m_0(p_b) = 1$ (see example 4), the control law would be same. But as differ from the first case all of them would be implementable. Obtained control law would drive subsystems to target marking while each channel are nonnegative, i.e., $m_f(p_a) = 0$, $m_f(p_b) = 0$. Hence, we would conclude that $\boldsymbol{m}_f(P^1)$ and $\boldsymbol{m}_f(P^2)$ are reachable from $\boldsymbol{m}_0(P^1)$ and $\boldsymbol{m}_0(P^2)$, simultaneously, while the final marking of each channel is at least $\gamma = 0$.

V. Case study

Let us consider the DcontPN sketched in Fig.2 (taken from [9]) which models two manufacturing processes containing assembly operations. The system can perform two types of products with two subsystems. These subsystems communicate through 6 channels. The set of input(output) channels of subsystem 1(subsystem 2) is $P^{2,1} = \{p_{c1}, p_{c3}, p_{c5}\}$, and the set of input(output) channels of subsystem 1(subsystem 2) is $P^{1,2} = \{p_{c2}, p_{c4}, p_{c6}\}$.

Let the initial marking of subsystem 1 be $\boldsymbol{m}_0(P^1) = [0\ 0\ 0\ 0\ 0\ 1\ 1\ 5\ 0\ 0\ 1\ 0\ 3]^T$, of subsystem 2 be $\boldsymbol{m}_0(P^2) = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 3\ 1\ 3]^T$, and $m_0(p_{c1}) = 1$, $m_0(p_{c2}) = 0$, $m_0(p_{c3}) = 1$, $m_0(p_{c4}) = 1$, $m_0(p_{c5}) = 1$, $m_0(p_{c6}) = 0$. Let the target markings of the subsystems be: $\boldsymbol{m}_f(P^1) = [1\ 0\ 0\ 0\ 0\ 1\ 0\ 5\ 0\ 1\ 0\ 0\ 1]^T$ and $\boldsymbol{m}_f(P^2) = [0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 3\ 1\ 2]^T$.

Let us execute the control algorithm 1. At the side of subsystem 1, $\bar{x}(t_1) = \bar{x}(t_7) = \bar{x}(t_8) = 1$, $\bar{x}(t_2) = \bar{x}(t_3) = \bar{x}(t_4) = \bar{x}(t_5) = \bar{x}(t_6) = \bar{x}(t_9) = 0$ are calculated, yielding $q^{REQ}_{p_{c1}} = 0$, $q^{REQ}_{p_{c3}} = 0$, $q^{REQ}_{p_{c5}} = -1$. At the side of subsystem 2, $\bar{x}(t_{10}) = \bar{x}(t_{11}) = \bar{x}(t_{12}) = \bar{x}(t_{14}) = \bar{x}(t_{15}) = 1$, $\bar{x}(t_{16}) = \bar{x}(t_{17}) = \bar{x}(t_{18}) = 0$ $\bar{x}(t_{13}) = 2$ are calculated, yielding $q^{REQ}_{p_{c2}} = 1$, $q^{REQ}_{p_{c4}} = 1$, $q^{REQ}_{p_{c6}} = 0$.

According to these values, $h_{p_{c2}} = -1$, $h_{p_{c4}} = -1$, $h_{p_{c6}} = 0$ is obtained at step 5 for the output channels of subsystem 1. Hence, it is necessary to recompute the flow integrals. With this re-computation the final values are: $x(t_1) = x(t_7) = x(t_8) = 2$, $x(t_2) = x(t_3) = x(t_4) = x(t_5) = x(t_6) = x(t_9) = 1$.

However, for the output buffers of subsystem 2 we have $h_{p_{c1}} = 1$, $h_{p_{c3}} = 1$, $h_{p_{c5}} = 1$. Hence it is not necessary to recalculate the flow integrals at this side. The final values of flow integrals are same as those of the first calculation. By using these final control law, $\boldsymbol{m}_f(P^1)$, $\boldsymbol{m}_f(P^2)$ can be reached and the final markings of channels are $m_f(p_{c1}) = 0$, $m_f(p_{c2}) = 0$, $m_f(p_{c3}) = 0$, $m_f(p_{c4}) = 0$, $m_f(p_{c5}) = 1$, $m_f(p_{c6}) = 1$.

VI. Conclusions

Distributed systems are composed of several subsystems that exchange data in order to obtain a given goal. This paper has focused on distributed systems modeled by continuous Petri nets, a relaxation of conventional Petri nets in which the integrality constraint in the firing of transitions is removed. Each subsystem is modeled as a subnet, and the communication among subsystems is achieved by means of places connecting subsystems.

In the framework of distributed continuous Petri nets, a reachability control problem has been considered. The approach developed here is based on the design of a local controller for each subsystem. The main difficulties that must be taken into account when dealing with the mentioned control problem are related to the coordination among local controllers and the possibility of reaching the target marking in every subsystem simultaneously. These difficulties appear even in apparently simple distributed systems consisting of only two subsystems. It is proved that, under certain assumptions on the system, the proposed algorithm for the controllers always yields an appropriate control law. Moreover, such an algorithm can be used to establish a necessary and sufficient condition for the reachability of the target marking in every subsystem simultaneously.

Appendix

The appendix is divided in three subsections. The first subsection introduces two technical lemmas, the second subsection contains the proof of Theorem 4.1, and the third subsection contains the proof of Theorem 4.2.

A. Preliminary Lemmas

Lemma 6.1: Let $\mathcal{N}$ be DcontPN satisfying assumptions (A1), (A2) and (A3). Then at step 5 of the algorithm, it holds that $max(h^1_{p_b}, h^2_{p_a}) \geq \gamma$, $\forall p_b \in P^{1,2}$ and $\forall p_a \in P^{2,1}$.

Proof: Since $\boldsymbol{m}_f$ is reachable from $\boldsymbol{m}_0$ (see Assumption (A2)) in the centralized system, then there exists $\boldsymbol{x}$ such that $\boldsymbol{m}_f = \boldsymbol{m}_0 + C \cdot \boldsymbol{x}$, $m_f(p_a) \geq \gamma$ $\forall p_a \in P^{2,1}$, $m_f(p_b) \geq \gamma$ $\forall p_b \in P^{1,2}$. Considering the particular structure $\boldsymbol{x}$ can be split into two vectors $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ such that :

$$\begin{array}{ll} \boldsymbol{m}_f(P^1) = \boldsymbol{m}_0(P^1) + C^1 \cdot \boldsymbol{x^1} & (a) \\ \boldsymbol{m}_f(P^2) = \boldsymbol{m}_0(P^2) + C^2 \cdot \boldsymbol{x^2} & (b) \end{array} \tag{6}$$

Markings of input and output channels are:

$$\begin{aligned} m_f(p_a) &= m_0(p_a) + \sum_{t \in {}^\bullet p_a} Post(p_a, t) \cdot x^2(t) \\ &\quad - \sum_{t \in p_a^\bullet} Pre(p_a, t) \cdot x^1(t) \geq \gamma, \quad \forall p_a \in P^{2,1} \\ m_f(p_b) &= m_0(p_b) + \sum_{t \in {}^\bullet p_b} Post(p_b, t) \cdot x^1(t) \\ &\quad - \sum_{t \in p_b{}^\bullet} Pre(p_b, t) \cdot x^2(t) \geq \gamma, \quad \forall p_b \in P^{1,2} \end{aligned} \tag{7}$$

In the algorithm, since $\boldsymbol{m}_f$ is reachable from $\boldsymbol{m}_0$, LPP (4) is feasible for both subsystems. Let $\bar{\boldsymbol{x}}^1$ be the solution for subsystem 1 and $\bar{\boldsymbol{x}}^2$ be the solution for subsystem 2. Given

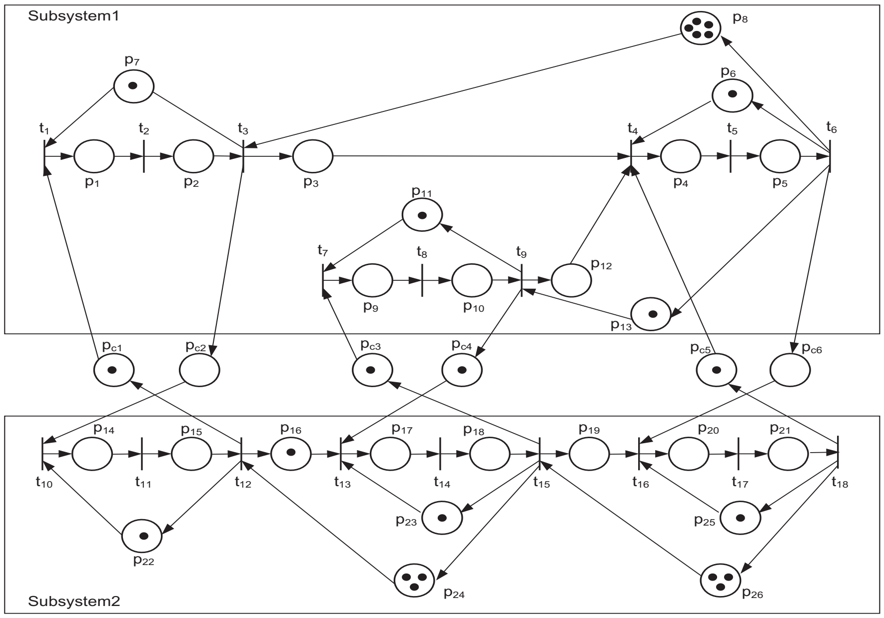

Fig. 2. A DcontPN modelling two manufacturing processes

that, both subsystems are MTS (there exists only one T-semiflow):

$$\begin{aligned} \boldsymbol{x}^1 = \bar{\boldsymbol{x}}^1 + \alpha^1 \cdot \boldsymbol{s}^1 \\ \boldsymbol{x}^2 = \bar{\boldsymbol{x}}^2 + \alpha^2 \cdot \boldsymbol{s}^2 \end{aligned} \tag{8}$$

where $\boldsymbol{s}^1$, $\boldsymbol{s}^2$ are the minimal T-semiflows of subsystems 1 and 2 and $\alpha^1, \alpha^2 \in \mathbb{R}^+$. By substituting (8) into (7):

$$\begin{aligned} m_f(p_a) &= m_0(p_a) + \sum_{t \in {}^\bullet p_a} Post(p_a, t) \cdot \bar{x}^2(t) \\ &+ \sum_{t \in {}^\bullet p_a} Post(p_a, t) \cdot \alpha^2 \cdot s^2(t) - \sum_{t \in p_a^\bullet} Pre(p_a, t) \cdot \bar{x}^1(t) \\ &- \sum_{t \in p_a^\bullet} Pre(p_a, t) \cdot \alpha^1 \cdot s^1(t) \geq \gamma, \quad \forall p_a \in P^{2,1} \\ m_f(p_b) &= m_0(p_b) + \sum_{t \in {}^\bullet p_b} Post(p_b, t) \cdot \bar{x}^1(t) \\ &+ \sum_{t \in {}^\bullet p_b} Post(p_b, t) \cdot \alpha^1 \cdot s^1(t) - \sum_{t \in p_b{}^\bullet} Pre(p_b, t) \cdot \bar{x}^2(t) \\ &- \sum_{t \in p_b^\bullet} Pre(p_b, t) \cdot \alpha^2 \cdot s^2(t) \geq \gamma, \quad \forall p_b \in P^{1,2} \end{aligned} \tag{9}$$

Because of the assumption (A3), the sum of $m_f(p_a)$ and $m_f(p_b)$ is simplified and written depending on $\bar{x}^1$ and $\bar{x}^2$:

$$\begin{aligned} m_f(p_a) + m_f(p_b) &= m_0(p_a) + \sum_{t \in {}^\bullet p_a} Post(p_a, t) \cdot \bar{x}^2(t) \\ &- \sum_{t \in p_a^\bullet} Pre(p_a, t) \cdot \bar{x}^1(t) + \sum_{t \in {}^\bullet p_b} Post(p_b, t) \cdot \bar{x}^1(t) \\ &+ m_0(p_b) - \sum_{t \in p_b{}^\bullet} Pre(p_b, t) \cdot \bar{x}^2(t) \geq 2 \cdot \gamma, \\ &\forall p_a \in P^{2,1}, \forall p_b \in P^{1,2} \end{aligned} \tag{10}$$

On the other hand, at step 4 of the Algorithm subsystem 1 receives the value of $r_{p_b}^{REQ} = \sum_{t \in p_b^\bullet} Pre(p_b, t) \cdot \bar{x}^2(t) - m_0(p_b)$, $\forall p_b \in P^{1,2}$ from subsystem 2 and subsystem 2 receives the value of $r_{p_a}^{REQ} = \sum_{t \in p_a^\bullet} Pre(p_a, t) \cdot \bar{x}^1(t) - m_0(p_a)$, $\forall p_a \in P^{2,1}$ from subsystem 1. At step 5 of the algorithm, how much tokens are put to the output channel by the present control law, which provides $\bar{x}^1$ and $\bar{x}^2$, are calculated, which are $h_{p_b}^1$, $\forall p_b \in P^{1,2}$ at the side of subsystem 1 and $h_{p_a}^2$, $\forall p_a \in P^{2,1}$ at the side of subsystem 2:

$$\begin{aligned} h_{p_b}^1 &= m_0(p_b) + \sum_{t \in {}^\bullet p_b} Post(p_b, t) \cdot \bar{x}^1(t) \\ &- \sum_{t \in p_b{}^\bullet} Pre(p_b, t) \cdot \bar{x}^2(t), \quad \forall p_b \in P^{1,2} \\ h_{p_a}^2 &= m_0(p_a) + \sum_{t \in {}^\bullet p_a} Post(p_a, t) \cdot \bar{x}^2(t) \\ &- \sum_{t \in p_a^\bullet} Pre(p_a, t) \cdot \bar{x}^1(t), \quad \forall p_a \in P^{2,1} \end{aligned} \tag{11}$$

In contrast to the lemma, assume there exist $p_a \in P^{2,1}$ and $p_b \in P^{1,2}$ such that $max(h_{pb}^1, h_{pa}^2) \leq \gamma$, then their sum has to be less than $2 \cdot \gamma$:

$$\begin{aligned} h_{p_b}^1 + h_{p_a}^2 &= m_0(p_b) + \sum_{t \in {}^\bullet p_b} Post(p_b, t) \cdot \bar{x}^1(t) \\ &- \sum_{t \in p_b{}^\bullet} Pre(p_b, t) \cdot \bar{x}^2(t) + \sum_{t \in {}^\bullet p_a} Post(p_a, t) \cdot \bar{x}^2(t) \\ &+ m_0(p_a) - \sum_{t \in p_a^\bullet} Pre(p_a, t) \cdot \bar{x}^1(t) < 2 \cdot \gamma \\ &p_a \in P^{2,1}, \; p_b \in P^{1,2} \end{aligned} \tag{12}$$

which is contrary to equation (10). ■

Lemma 6.2: Let $\mathcal{N}$ be DcontPN satisfying assumptions (A1), (A2) and (A3). Then, if $h^1_{p_b} < \gamma$ for at least one $p_b \in P^{1,2}$ and $h^2_{p_a} \geq \gamma, \ \ \forall p_a \in P^{2,1}$, then Algorithm 1 computes a control law that:

- drives the subsystems from $\boldsymbol{m}_0(P^1)$ and $\boldsymbol{m}_0(P^2)$ to target markings $\boldsymbol{m}_f(P^1)$ and $\boldsymbol{m}_f(P^2)$, simultaneously.
- the final markings of channels satisfy $m_f(p) \geq \gamma \ \forall p \in P^{1,2} \cup P^{2,1}$

Proof: Since $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are assumed to be reachable in the centralized system (assumption (A2)); LPP (4) is feasible for both subsystem 1 and subsystem 2 yielding $\bar{\boldsymbol{x}}^1$ and $\bar{\boldsymbol{x}}^2$, that is:

$$\begin{aligned} \boldsymbol{m}_f(P^1) &= \boldsymbol{m}_0(P^1) + \boldsymbol{C}^1 \cdot \bar{\boldsymbol{x}}^1 \quad (a) \\ \boldsymbol{m}_f(P^2) &= \boldsymbol{m}_0(P^2) + \boldsymbol{C}^2 \cdot \bar{\boldsymbol{x}}^2 \quad (b) \end{aligned} \tag{13}$$

At step 4, subsystem 1 receives $r^{REQ}_{p_b} = \sum_{t \in p_b^\bullet} Pre(p_b,t) \cdot \bar{x}^2(t) - m_0(p_b), \ \forall p_b \in P^{1,2}$ from subsystem 2 and subsystem 2 receives the value of $r^{REQ}_{p_a} = \sum_{t \in p_a^\bullet} Pre(p_a,t) \cdot \bar{x}^1(t) - m_0(p_a), \ \forall p_a \in P^{2,1}$ from subsystem 1.

At the side of subsystem 1: Since $h^1_{p_b} < \gamma$ for at least one $p_b \in P^{1,2}$, step 6 will be performed, that is LPP (5) will be solved at the side of subsystem 1 to obtain new flow integral vector, $\boldsymbol{x}^1$. LPP (5) is feasible because of assumption (A2). So, $\boldsymbol{x}^1$ drives the system from $\boldsymbol{m}_0(P^1)$ to $\boldsymbol{m}_f(P^1)$:

$$\boldsymbol{m}_f(P^1) - \boldsymbol{m}_0(P^1) = \boldsymbol{C}^1 \cdot \boldsymbol{x}^1 \tag{14}$$

At the side of subsystem 2: Since $h^2_{p_a} \geq 0, \ \ \forall p_b \in P^{2,1}$ step 6 is not performed, the final flow integral vector for subsystem 2 is:

$$\boldsymbol{x}^2 = \bar{\boldsymbol{x}}^2 \tag{15}$$

Since $\boldsymbol{x}^2$ is the solution of LPP (4), it drives subsystem 2 from $\boldsymbol{m}_0(P^2)$ to $\boldsymbol{m}_f(P^2)$.

So far, we explained that the control law providing final flow integral vectors drive the subsystems from initial states $\boldsymbol{m}_0(P^1)$ and $\boldsymbol{m}_0(P^2)$ to target states $\boldsymbol{m}_f(P^1)$ and $\boldsymbol{m}_f(P^2)$, simultaneously.

In the following, we will show the control law providing final flow integral vectors $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ ensure $m_f(p_a) \geq \gamma \ \ \forall p_a \in P^{2,1}$ and $m_f(p_b) \geq \gamma \ \ \forall p_b \in P^{1,2}$.

$\bar{\boldsymbol{x}}^1$ is the solution of LPP (4) and $\boldsymbol{x}^1$ is the solution of LPP (5) for subsystem 1. Given that subsystems are MTS (there exists only one T-semiflow):

$$\boldsymbol{x}^1 = \bar{\boldsymbol{x}}^1 + \alpha^1 \cdot \boldsymbol{s}^1 \tag{16}$$

where $\boldsymbol{s}^1$ is minimal T-semiflow of subsystem 1, $\alpha^1 \in \mathbb{R}^+$. According to LPP (5) in step 6 of the algorithm, the constraint

$$\begin{aligned} \sum_{t \in {}^\bullet p_b} Post(p_b,t) \cdot x^1(t) \geq \sum_{t \in p_b^\bullet} Pre(p_b,t) \cdot \bar{x}^2(t) \\ - m_0(p_b) + \gamma, \quad \forall p_b \in P^{1,2} \end{aligned} \tag{17}$$

must be satisfied. Subtracting $\sum_{t \in {}^\bullet p_b} Post(p_b,t) \cdot \bar{x}^1(t)$ from both side of this inequality yields:

$$\begin{aligned} &\sum_{t \in {}^\bullet p_b} Post(p_b,t) \cdot x^1(t) - \sum_{t \in {}^\bullet p_b} Post(p_b,t) \cdot \bar{x}^1(t) \geq \\ &\sum_{t \in p_b^\bullet} Pre(p_b,t) \cdot \bar{x}^2(t) - \sum_{t \in {}^\bullet p_b} Post(p_b,t) \cdot \bar{x}^1(t) \\ &\qquad\qquad - m_0(p_b) + \gamma \quad \forall p_b \in P^{1,2} \end{aligned} \tag{18}$$

Then, by taking (16) and (18) into the consideration, LPP (5) yields α^1 such that:

$$\begin{aligned} &\sum_{t \in {}^\bullet p_b} Post(p_b,t) \cdot \alpha^1 \cdot s^1(t) \\ &= \max_{p_b \in P^{1,2}} \{ \sum_{t \in p_b^\bullet} Pre(p_b,t) \cdot \bar{x}^2(t) - \sum_{t \in {}^\bullet p_b} Post(p_b,t) \cdot \bar{x}^1(t) \\ &\qquad\qquad - m_0(p_b) + \gamma \} \end{aligned} \tag{19}$$

Let $p_x \in P^{1,2}$ be the place which gives that maksimum value. Then,

$$\begin{aligned} &\sum_{t \in {}^\bullet p_b} Post(p_b,t) \cdot \alpha^1 \cdot s^1(t) \\ &= \sum_{t \in p_x^\bullet} Pre(p_x,t) \cdot \bar{x}^2(t) - \sum_{t \in {}^\bullet p_x} Post(p_x,t) \cdot \bar{x}^1(t) \\ &\qquad - m_0(p_x) + \gamma \geq 0, \quad p_x, \ \forall p_b \in P^{1,2} \end{aligned} \tag{20}$$

By using the flow integral values given in (16) and (15), marking of an input channel of subsystem 1 is:

$$\begin{aligned} m_f(p_a) &= m_0(p_a) + \sum_{t \in {}^\bullet p_a} Post(p_a,t) \cdot x^2(t) \\ &\quad - \sum_{t \in p_a^\bullet} Pre(p_a,t) \cdot x^1(t) \\ &= m_0(p_a) + \sum_{t \in {}^\bullet p_a} Post(p_a,t) \cdot \bar{x}^2(t) \\ &\quad - \sum_{t \in p_a^\bullet} Pre(p_a,t) \cdot \bar{x}^1(t) - \sum_{t \in p_a^\bullet} Pre(p_a,t) \cdot \alpha^1 \cdot s^1(t) \\ &\qquad \forall p_a \in P^{2,1} \end{aligned} \tag{21}$$

According to assumption (A3), $\sum_{t \in p_a^\bullet} Pre(p_a,t) \cdot \alpha^1 \cdot s^1(t) = \sum_{t \in {}^\bullet p_b} Post(p_b,t) \cdot \alpha^1 \cdot s^1(t)$. Then $m_f(p_a)$ is:

$$\begin{aligned} m_f(p_a) &= m_0(p_a) + \sum_{t \in {}^\bullet p_a} Post(p_a,t) \cdot \bar{x}^2(t) \\ &\quad - \sum_{t \in p_a^\bullet} Pre(p_a,t) \cdot \bar{x}^1(t) - \sum_{t \in p_b^\bullet} Pre(p_x,t) \cdot \bar{x}^2(t) \\ &\quad + \sum_{t \in {}^\bullet p_b} Post(p_x,t) \cdot \bar{x}^1(t) + m_0(p_x) - \gamma \\ &\qquad \forall p_a \in P^{2,1} \quad p_b \in P^{1,2} \end{aligned}$$

which can not be less than γ according to equation (10). Similarly, by flow integral values given in (16) and (15), marking of an input channel of subsystem 2 is:

$$\begin{aligned} m_f(p_b) &= m_0(p_b) + \sum_{t\in {}^\bullet p_b} Post(p_b,t)\cdot x^1(t) \\ &\quad - \sum_{t\in p_b^\bullet} Pre(p_b,t)\cdot x^2(t) \\ &= m_0(p_b) + \sum_{t\in {}^\bullet p_b} Post(p_b,t)\cdot \bar{x}^1(t) \\ &+ \sum_{t\in {}^\bullet p_b} Post(p_b,t)\cdot \alpha^1\cdot s^1(t) - \sum_{t\in p_b^\bullet} Pre(p_b,t)\cdot \bar{x}^2(t), \\ &\qquad \forall p_b \in P^{1,2} \end{aligned} \tag{22}$$

According to (20), $\sum_{t\in {}^\bullet p_b} Post(p_b,t)\cdot\alpha^1\cdot s^1(t) = \sum_{t\in p_x{}^\bullet} Pre(p_x,t)\cdot\bar{x}^2(t) - \sum_{t\in {}^\bullet p_x} Post(p_x,t)\cdot\bar{x}^1(t) - m_0(p_x)+\gamma$. Hence $m_f(p_b)$ is:

$$\begin{aligned} m_f(p_b) &= m_0(p_b) + \sum_{t\in {}^\bullet p_b} Post(p_b,t)\cdot \bar{x}^1(t) \\ &+ \sum_{t\in p_x{}^\bullet} Pre(p_x,t)\cdot \bar{x}^2(t) - \sum_{t\in {}^\bullet p_x} Post(p_x,t)\cdot \bar{x}^1(t) \\ &- m_0(p_x) - \sum_{t\in p_b^\bullet} Pre(p_b,t)\cdot\bar{x}^2(t) + \gamma, \\ &\qquad p_x,\ \forall p_b\in P^{1,2} \end{aligned}$$

which can not be less than γ, because of the definition of place p_x in (20) such that:

$$\begin{aligned} &\sum_{t\in p_x{}^\bullet} Pre(p_x,t)\cdot \bar{x}^2(t) - \sum_{t\in {}^\bullet p_x} Post(p_x,t)\cdot \bar{x}^1(t) \\ &\quad - m_0(p_x) \geq \sum_{t\in p_b{}^\bullet} Pre(p_b,t)\cdot\bar{x}^2(t) - m_0(p_b) \\ &\quad - \sum_{t\in {}^\bullet p_b} Post(p_b,t)\cdot\bar{x}^1(t), \quad p_x,\ \forall p_b\in P^{1,2} \end{aligned} \tag{23}$$

■

B. Proof of Theorem 4.1

Proof: There are four possible cases for the number of tokens which are put the output channels by the control already calculated:

(a) $h^1_{p_b} < \gamma$ for at least one $p_b \in P^{1,2}$ and $h^2_{p_a} < \gamma$ for at least one $p_a \in P^{2,1}$: This case does not appear according to Lemma 6.1.

(b) $h^1_{p_b} \geq \gamma,\ \ \forall p_b \in P^{1,2}$ and $h^2_{p_a} \geq \gamma,\ \ \forall p_a \in P^{2,1}$: Since $\boldsymbol{m_f}$ is reachable from $\boldsymbol{m}_0$ in the centralized system LPP (4) is feasible and produces $\bar{x}^1$ at the side of subsystem 1 and $\bar{x}^2$ at the side of subsystem 2. By the control law providing $\bar{\boldsymbol{x}}^1$ and $\bar{\boldsymbol{x}}^2$, subsystem 1 reaches $\boldsymbol{m}_f(P^1)$ from $\boldsymbol{m}_0(P^1)$ and subsystem 2 reaches $\boldsymbol{m}_f(P^2)$ from $\boldsymbol{m}_0(P^2)$. Because $h^1_{p_b} \geq \gamma,\ \ \forall p_b \in P^{1,2}$ and $h^2_{p_a} \geq \gamma,\ \ \forall p_a \in P^{2,1}$ obtained by these flow integrals, this control yields $m_f(p_a) \geq \gamma\ \ \forall p_a \in P^{2,1}$ and $m_f(p_b) \geq \gamma\ \ \forall p_b \in P^{1,2}$.

(c) $h^1_{p_b} < \gamma$ for at least one $p_b \in P^{1,2}$ and $h^2_{p_a} \geq \gamma,\ \ \forall p_a \in P^{2,1}$: According to Lemma 6.2 algorithms compute the corresponding control law.

(d) $h^2_{p_a} < \gamma$ for at least one $p_a \in P^{2,1}$ and $h^1_{p_b} \geq \gamma,\ \ \forall p_b \in P^{1,2}$: Proof of this case is similar to the case (c). ■

C. Proof of Theorem 4.2

Proof: If ⇒This part is same as the proof of Theorem 4.2. *Only If* ⇒ Let we require min number of γ tokens in communication channels. Then reachability of $\boldsymbol{m}_f$ for overall system, i.e., $\boldsymbol{m_f} = \boldsymbol{m}_0 + \boldsymbol{C}\cdot\boldsymbol{\sigma}$, can be expanded to the following equations, where $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ are final flow integral values obtained by the execution of the algorithm:

$$\begin{aligned} &\boldsymbol{m_f}(P^1) = \boldsymbol{m}_0(P^1) + \boldsymbol{C}^1\cdot\boldsymbol{x}^1 && (a) \\ &m_f(p) = m_0(p) + \sum_{t\in {}^\bullet p} Post(p,t)\cdot x^2(t) \\ &\quad - \sum_{t\in p\bullet} Pre(p,t)\cdot x^1(t) \geq \gamma, \qquad \forall p\in P^{1,2} && (b) \\ &\boldsymbol{m_f}(P^2) = \boldsymbol{m}_0(P^2) + \boldsymbol{C}^2\cdot\boldsymbol{x}^2 && (c) \\ &m_f(p) = m_0(p) + \sum_{t\in {}^\bullet p} Post(p,t)\cdot x^1(t) \\ &\quad - \sum_{t\in p\bullet} Pre(p,t)\cdot x^2(t) \geq \gamma, \qquad \forall p\in P^{2,1} && (d) \end{aligned}$$

If (a) or (b) is not satisfied constraints of LPP (5) are not satisfied at the side of subsystem 1, if (c) or (d) is not satisfied constraints of LPP (5) are not satisfied at the side of subsystem 2. Hence the final flow integral values do not yield a feasible control law. ■

References

[1] L. Giacomani B. Bordbar and D.J. Holding. Design of distributed manufacturing systems using UML and Petri nets. In *Proc. of 6th IFAC Workshop on Algorithms and Architectures for Real-Time Control*, pages 91–96, Mallorca, Spain, May 2000.

[2] R. David and H.Alla. Continuous Petri nets. In *8th European Workshop on Application and Theory of Petri Nets*, Zaragoza, Spain, 1987.

[3] R. David and H.Alla. *Autonomous and timed continuous Petri nets*. Springer, Berlin, 2005.

[4] H.Alla and R.David. Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers*, 8:159–188, 1998.

[5] P. Krishnan. Distributed timed automata. 2000.

[6] 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.

[7] 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.

[8] T. Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77 No:4:541–580, 1989.

[9] J.M. Proth. Petri nets for modelling and evaluating deterministic and stochastic manufacturing systems. In *PNPM '97: Proceedings of the 6th International Workshop on Petri Nets and Performance Models*, pages 3192–3197, Washington, DC, USA, 1997.

[10] J.l.V.Salcedo R. P. Moreno, D. Tardioli. Distributed implementation of discrete event control systems based on Petri nets. In *ISIE08:IEEE International Symposium on Industrial Electronics*, pages 1738–1745, June 2008.

[11] 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.