A Final Marking Planning Method for Join Free Timed Continuous Petri nets

Hanife Apaydın Özkan and Aydın Aybar

Abstract— **In this paper, an online control method is developed and corresponding algorithm is proposed for driving Join Free continuous Petri net from its initial marking, to target marking through a linear trajectory by minimizing the time. Then, the control problem in which some components of the target marking are not specified is considered and developed control method is used for that case.**

I. INTRODUCTION

Discrete Petri Nets (PNs) are powerful graphical and mathematical tools for modeling, analysis and synthesis of *Discrete Event System*s (DESs) [1], [2]. The distributed state or marking of a PN is given by a vector of natural numbers which represent the number of tokens in each place. This is a significant advantage with respect to other formalisms such as automata, where the state space is a symbolic unstructured set.

Like in most modeling formalisms for DESs, PNs suffer from the so called *state explosion* which leads to an exponential growth of the size of state space with respect to the size of the system and population of initial state. Some relaxation techniques are studied to overcome this difficulty and to reduce the computational complexity of the analysis and synthesis of PNs (i.e. decomposition techniques, Lagrangian relaxations, fluidification and the others [3], [4], [5], [6]). *Fluidification* may be very useful relaxation technique when applied to highly populated systems.

For PNs, fluidification was introduced in [3], [7] aiming at giving fluid (continuous) approximation of original PN in the sense of behaviours and properties, and these models are called *continuous Petri nets*. The idea is to try to overcome, at least partially, the potentially very high computational complexity arising in many practical situations.

Different techniques have been proposed for control of continuous Petri nets in the literature [8], [9], [10], [11], [12]. Steady state optimal control of continuous Petri nets was studied in [13] where it is shown that, the optimal steady state control problem of continuous Petri net system can be solved by means of *Linear Programming Problem* (LPP) in the case that all transitions are controllable and the objective function is linear. For the problem of reaching a given steady state from an initial marking, implicit and explicit *Model Predictive Control* (MPC) methods minimizing a certain performance index are proposed in [8]. The step tracking problem, i.e. design of control laws to drive the system states to target references was considered in [10] and a Lyapunov-function-based dynamic control algorithm was proposed for the problem. That method requires solving a *BiLinear Programming Problem* (BLP) for the computation of intermediate states. In [12] an efficient heuristics for minimum time control of continuous Petri nets, which aims at driving the system from an initial state to a target one by minimizing the time of a piecewise linear trajectory is developed.

In some control problems, final states of some places may not be specified, while that others are specified. Because reaching desired final states of other places in minimum time is more important then the unspecified final states. Accordingly, corresponding components of the target state are not specified in the control problem. In this paper, this problem is considered for continuous Petri nets for the first time in the literature.

For calculating the value of unspecified components under the objective of time minimization, we developed an online control strategy. This strategy focuses on timed continuous Petri nets (contPN) without synchronizations called Join Free contPN and drives JF contPN to a specified target state through a linear trajectory by means of LPP. This method proposes an online algorithm and requests solving BLP for calculating unspecified components.

The remainder of the paper is organized as follows. Section 2 briefly introduces the required concepts of contPN systems and introduces the formulation of applied control. A new control scheme for JF contPN is given in Section 3. In Section 4, a method for calculating unspecified components of target marking under the objective of time minimization is adressed. Finally, some conclusions and future directions are drawn in Section 5.

II. BASIC CONCEPTS AND NOTATION

We assume that the reader is familiar with Petri nets. A continuous Petri net system is a pair $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ where $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \ \boldsymbol{Post} \rangle$ is a net structure where $P = \{p_1, \ p_2, \ ...p_{|P|}\}$ and $T = \{t_1, \ t_2, ... \ t_{|T|}\}$ are the sets of places and transitions, respectively; $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{N}^{|P| \times |T|}$ are pre and post matrices connecting places and transitions; $\boldsymbol{m}_0 \in \mathbb{R}_{\geq 0}^{|P|}$ is initial marking (state).

For a place $p_i \in P$ and a transition $t_j \in T$, Pre_{ij} and $Post_{ij}$ 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.

Hanife Apaydın Özkan and Aydın Aybar are with Department of Electrical and Electronics Engineering, Anadolu University, Eskişehir, Turkey hapaydin1,aaybar@anadolu.edu.tr

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

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

which represents the maximum amount in which t_j can fire. An enabled transition t_j can fire in any real amount α, with $0 < \alpha \le enab(t_j, \boldsymbol{m})$ 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 $C_{\cdot j}$ is its j^{th} column. If $\boldsymbol{m}$ is reachable from $\boldsymbol{m}_0$ through a finite sequence $\boldsymbol{\sigma}$, the state (or fundamental) equation is satisfied: $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, where $\boldsymbol{\sigma} \in \mathbb{R}_{\ge 0}^{|T|}$ is the firing count vector, i.e., σ_j is the cumulative amount of firings of t_j in the sequence σ. The set of reachable markings from $\boldsymbol{m}_0$ is denoted by $RS(\mathcal{N}, \boldsymbol{m}_0)$.

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

A timed continuous Petri net (contPN) is a continuous Petri net together with a vector $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{|T|}$ where λ_j is the firing rate of t_j. As in untimed continuous Petri nets state equation summarizes the way the marking evolves along time. The state equation of contPN has an explicit dependence on time $\boldsymbol{m}(\tau) = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}(\tau)$ where τ is global time. But, in continuous systems, the marking is continuously changing, so we may consider the derivative of $\boldsymbol{m}$ with respect to time. This way, $\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \dot{\boldsymbol{\sigma}}(\tau)$ is obtained. Here, $\dot{\boldsymbol{\sigma}}(\tau)$ is flow throgh transitions and it is denoted by $\boldsymbol{f}(\tau) = \dot{\boldsymbol{\sigma}}(\tau)$. Hence, the state equation is

$$\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \boldsymbol{f}(\tau) \tag{2}$$

Different semantics have been defined for continuous timed transitions [14], [3]. Infinite server semantics is considered in this paper. Under this semantics, the flow of transition t_j is the product of firing rate, λ_j, and enabling of transition $enab(t_j, \boldsymbol{m}(\tau))$:

$$f_j(\tau) = \lambda_j \cdot enab(t_j, \boldsymbol{m}(\tau)) = \lambda_j \cdot \min_{p_i \in^{\bullet} t_j} \left\{ \frac{m_i(\tau)}{Pre_{ij}} \right\} \tag{3}$$

For the sake of simplicity τ is omitted in the rest of the paper.

We consider Join Free contPNs (JF contPNs) which satisfy $|^{\bullet}t_j|\ \ \forall j \in \{0...|T|\}$. Let us define matrix $\boldsymbol{\Pi} \in \mathbb{R}_{\ge 0}^{|T| \times |P|}$ as:

$$\Pi_{ji} = \begin{cases} \frac{1}{Pre_{ij}}, & if \quad Pre_{ij} \ne 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The state equation of uncontrolled JF contPN is as follow:

$$\dot{\boldsymbol{m}} = \boldsymbol{C} \cdot \boldsymbol{f} = \boldsymbol{C} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi} \cdot \boldsymbol{m} \tag{5}$$

where $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, ...\lambda_{|T|}\}$. Reachability set of JF contPN is denoted by $RS(\mathcal{N},\ \boldsymbol{\lambda},\ \boldsymbol{m}_0)$. Given that the continuous Petri nets that we are considering are Join Free and every transition is fireable, the set of reachable markings is equal to the solutions of the state equation $RS(\mathcal{N},\ \boldsymbol{\lambda},\ \boldsymbol{m}_0) = \{\boldsymbol{m} \mid \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}\}, \boldsymbol{\sigma} \in \mathbb{R}_{\ge 0}^{|T|}$.

A. Control Scheme

Now let us introduce control concept that we consider in this paper. In contPN, a transition is associated in general to a machine and this machine can not work faster then its maximum firing rate, the only control action we consider is to brake it down. In other words, we assume that the only action that can be applied to contPN is to reduce the flow of transitions [15]. If a transition can be controlled (its flow can be reduced or even stopped), we will say that it is a *controllable* transition [13]. In this paper, it is assumed that all transitions are controllable.

The controlled flow, $\boldsymbol{w}$, of a contPN is defined as $\boldsymbol{w}(\tau) = \boldsymbol{f}(\tau) - \boldsymbol{u}(\tau)$, with $\boldsymbol{0} \le \boldsymbol{u}(\tau) \le \boldsymbol{f}(\tau)$, where $\boldsymbol{f}$ is the flow of the uncontrolled system, i.e., defined as in (5), 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 a JF contPN system in which all transitions are controllable is ruled by the following system:

$$\begin{array}{l} \dot{\boldsymbol{m}} = \boldsymbol{C} \cdot [\boldsymbol{f}\ -\ \boldsymbol{u}] = \boldsymbol{C} \cdot \boldsymbol{w} \\ \boldsymbol{0} \le \boldsymbol{u} \le \boldsymbol{f} \end{array} \tag{6}$$

The constraint $\boldsymbol{0} \le \boldsymbol{u} \le \boldsymbol{f}$ can be rewritten as $\boldsymbol{0} \le \boldsymbol{f} - \boldsymbol{u} \le \boldsymbol{f}$. From the definition, $\boldsymbol{w} = \boldsymbol{f} - \boldsymbol{u}$, the constraint can be expressed as:

$$\boldsymbol{0}\ \le\ \boldsymbol{w}\ \le\ \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi} \cdot \boldsymbol{m} \tag{7}$$

The following sections focus on the control problem for JF subclass in the case that some components of target markings are not specified. In Section 3, an online control method is developed and corresponding algorithm is proposed for driving JF contPN from its initial marking, $\boldsymbol{m}_0$, to target marking, $\boldsymbol{m}_f$, through a linear trajectory by minimizing the time. Section 4 makes use of developed method for the control problem in which some components of target marking are not specified. We assume that $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are strictly positive. The assumption that $\boldsymbol{m}_0$ is positive ensures that the system can move at $\tau = 0$ in the direction of $\boldsymbol{m}_f$ [16]; the assumption that $\boldsymbol{m}_f$ is positive ensures that $\boldsymbol{m}_f$ can be reached in finite time [13].

III. A Control Method for JF contPNs

In this section, an online control method that drives the system from the initial marking $\boldsymbol{m}_0$ to a desired target marking $\boldsymbol{m}_f$ through a linear trajectory will be introduced.

Our procedure consists of using discrete time representation of the system, and calculating control input at each sampling instant by using the maximum flows of transitions at $\boldsymbol{m}_0$ in the direction to $\boldsymbol{m}_f$ and maximum flows of transitions at $\boldsymbol{m}_f$ in the direction from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$.

Maximum flow of transitions at $\boldsymbol{m}_0$ in the direction to $\boldsymbol{m}_f$ is denoted by $\boldsymbol{w}_0$ and it is calculated by the following

LPP, where $\boldsymbol{s}_0 = \boldsymbol{w}_0 \cdot \tau_0$:

$$\begin{array}{lll} \min\limits_{\boldsymbol{s}_0} & \tau_0 & \\ \text{s.t.} & \boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{s}_0 & (a) \\ & 0 \leq s_{0j} \leq \lambda_j \cdot \Pi_{ji} \cdot m_{0i} \cdot \tau_0 & \\ & \forall\, j \in \{1,..,|T|\} \text{ where } i \text{ satisfies } \Pi_{ji} \neq 0 & (b) \end{array} \quad (8)$$

The equations correspond to: (a) the straight line connecting $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$, (b) flow constraints at $\boldsymbol{m}_0$. Notice that (b) is a linear constraint because m_{0i} and m_{f_i} are known $\forall i \in \{1, 2, .., |P|\}$.

Maximum flow of transitions at $\boldsymbol{m}_f$ in the direction from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$ is denoted by $\boldsymbol{w}_f$ and it is calculated by the following LPP, where $\boldsymbol{s}_f = \boldsymbol{w}_f \cdot \tau_f$:

$$\begin{array}{lll} \min\limits_{\boldsymbol{s}_f} & \tau_f & \\ \text{s.t.} & \boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{s}_f & (a) \\ & 0 \leq s_{f_j} \leq \lambda_j \cdot \Pi_{ji} \cdot m_{f_i} \cdot \tau_f & \\ & \forall\, j \in \{1,..,|T|\} \text{ where } i \text{ satisfies } \Pi_{ji} \neq 0 & (b) \end{array} \quad (9)$$

The equations correspond to: (a) the straight line connecting $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$, (b) flow constraints at $\boldsymbol{m}_f$. Notice that (b) is a linear constraint because m_{0i} and m_{f_i} are known $\forall i \in \{1, 2, .., |P|\}$.

Proposition: Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a contPN system with $\boldsymbol{m}_0 > \boldsymbol{0}$. If $\boldsymbol{m}_f$ belongs to $RS(\mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0)$ and $\boldsymbol{m}_f > 0$, then LPPs in (8) and (9) are feasible.

Proof: Since $\boldsymbol{m}_f$ is a reachable marking, then there exists $\boldsymbol{s}$ such that the state equations (8)(a) and (9)(a) are satisfied. By taking τ_0 and τ_f sufficiently large (8)(b) and (9)(b) can be satisfied since $\lambda_j \cdot \Pi_{ji} \cdot m_{0i} > 0$ and $\lambda_j \cdot \Pi_{ji} \cdot m_{f_i} > 0$.□

Since it is assumed that, $\boldsymbol{m}_0 > \boldsymbol{0}$ and $\boldsymbol{m}_f > \boldsymbol{0}$, then the linear trajectory from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$ can be followed by the system [16]. At each marking $\boldsymbol{m}$ on the line connecting $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$ (i.e. $\boldsymbol{m} = \alpha \cdot \boldsymbol{m}_0 + (1-\alpha) \cdot \boldsymbol{m}_f, \quad \alpha \in [0\ 1]$), the maximum flow in the direction to $\boldsymbol{m}_f$ is calculated by

$$\boldsymbol{w} = \alpha \cdot \boldsymbol{w}_0 + (1-\alpha) \cdot \boldsymbol{w}_f \quad (10)$$

Corresponding control action is calculated easily by $\boldsymbol{u} = \boldsymbol{f} - \boldsymbol{w}$ where $\boldsymbol{f}$ and $\boldsymbol{w}$ are controlled and uncontrolled flow vectors, respectively.

In order to calculate control inputs to drive the system from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$ through a linear trajectory and drive the system to $\boldsymbol{m}_f$ by using the calculated control we propose to use discrete-time representation of contPN. The discrete-time representation of the continuous-time system (6) is given by:

$$\begin{array}{l} \boldsymbol{m}[k+1] = \boldsymbol{m}[k] + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w}[k] \\ 0 \leq \boldsymbol{w}[k] \leq \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi} \cdot \boldsymbol{m}[k] \end{array} \quad (11)$$

Here Θ is the sampling period ($\tau = k \cdot \Theta$) and $\boldsymbol{m}[k]$ is the marking at step k, i.e., at time $k \cdot \Theta$. The sampling period should be small enough to avoid to reach negative markings. Let us consider a place p_i with $p_i^{\bullet} = \{t_1, t_2, ..., t_j\}$ and $m[k]_i > 0$. Then state equation can be written as $m[k+1]_i = m[k]_i + \Theta \cdot \boldsymbol{C}(i,:) \cdot \boldsymbol{w}[k] \geq m[k]_i - \Theta \cdot (\lambda_1 + \lambda_2 + ... + \lambda_j) \cdot m[k]_i = m[k]_i \cdot (1 - \sum\limits_{t_j \in p^{\bullet}} \lambda_j \cdot \Theta) \geq 0$. Hence, if Θ is chosen

Algorithm 1

Input: $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle, \quad \boldsymbol{m}_f, \quad \Theta$
Step 1) Solve LPP in (8) and LPP in (9)
Step 2) k=0
Step 3) If $\boldsymbol{m}[k] \neq \boldsymbol{m}_f$
Step 4) Measure

$$\boldsymbol{m}[k+1] = \boldsymbol{m}[k] + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w}[k] \quad (13)$$

Step 5) Calculate corresponding α

$$\alpha = \frac{\boldsymbol{m}[k+1]_1 - \boldsymbol{m}_{f_1}}{\boldsymbol{m}_{0_1} - \boldsymbol{m}_{f_1}} \quad (14)$$

Step 6) If $\alpha > 1$

$$\Theta = \frac{\boldsymbol{m}_{f_1} - \boldsymbol{m}[k]_1}{\boldsymbol{C}(1,:) \cdot \boldsymbol{w}[k]} \quad (15)$$

Step 7) Advance one step

$$\boldsymbol{m}[k+1] = \boldsymbol{m}[k] + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w}[k] \quad (16)$$

Step 8) Calculate corresponding $\boldsymbol{w}[k+1]$

$$\boldsymbol{w}[k+1] = \alpha \cdot \boldsymbol{w}_0 + (1-\alpha) \cdot \boldsymbol{w}_f \quad (17)$$

Step 9) $k = k+1$

such that

$$\sum_{t_j \in p^{\bullet}} \lambda_j \cdot \Theta < 1 \quad (12)$$

then any marking reachable from $\boldsymbol{m}_0 = \boldsymbol{m}[0] \geq 0$ is nonnegative [8].

In order to calculate controlled flows and drive the system to $\boldsymbol{m}_f$, Algorithm 1 is developed. In this algorithm, controlled flow is calculated at each sampling instant by using the fact given in (10). Here, the controlled flows $\boldsymbol{w}[0] = \boldsymbol{w}_0$ and $\boldsymbol{w}_f$ are obtained by solving LPPs in (8) and (9), respectively. Then, at the first iteration, $\boldsymbol{m}[1]$, α and $\boldsymbol{w}[1]$ (by using obtained α) are calculated. And obtained controlled flow is realized. At the next iteration, $\boldsymbol{m}[2]$, α and $\boldsymbol{w}[2]$ are calculated by similar way. This procedure is repeated until $\boldsymbol{m}_f$ is reached. During the execution of the algorithm, if α is obtained as bigger than 1, that is $\boldsymbol{m}_f$ is passed at the current iteration, Θ is recalculated to reach $\boldsymbol{m}_f$ accurately. We developed MATLAB program for Algorithm 1. This program is implemented on a PC with Intel(R) Core(TM) 2CPU T5600 @ 1.83GHz, 2.00 GB of RAM.

Example 1: Let us consider JF contPN in Fig. 1 with $\boldsymbol{\lambda} = [1\ 1\ 1\ 1\ 1\ 1]^T$. The only minimal P-semiflow is $\boldsymbol{y} = [2\ 2\ 1\ 1\ 1]$ and there are two minimal T-semiflows $\boldsymbol{x}^1 = [1\ 1\ 0\ 0\ 0\ 0]^T$ and $\boldsymbol{x}^2 = [0\ 0\ 1\ 1\ 1\ 1]^T$. Our aim is to drive the system from $\boldsymbol{m}_0 = [13\ 3\ 4\ 4\ 5]^T$ to a final state $\boldsymbol{m}_f = [10\ 6\ 6\ 3\ 2]^T$ by using the proposed control method. The system dynamics can be described as follows:

$$\begin{array}{l} \dot{m}_1 = m_2 + \frac{1}{2} \cdot m_5 - m_1 - m_1 \\ \dot{m}_2 = m_1 - m_2 \\ \dot{m}_3 = m_1 - m_3 \\ \dot{m}_4 = m_1 - m_4 \\ \dot{m}_5 = m_3 + m_4 - m_5 \end{array} \quad (18)$$

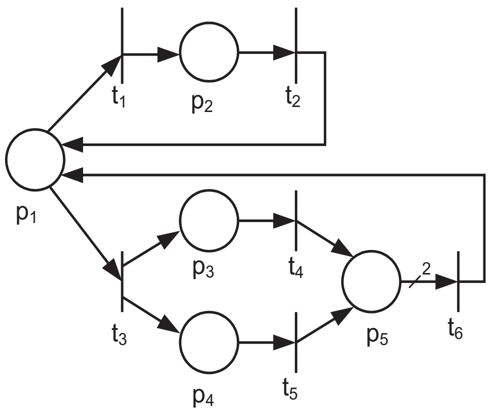

Fig. 1: A JF Petri net [9]

At $\boldsymbol{m}_0$, solving LPP (8) yields $\boldsymbol{w}_0 = [5.8\ 2.05\ 2.5\ 0\ 1.25\ 2.5]^T$ $\tau_0 = 0.8$ t.u. At $\boldsymbol{m}_f$, solving LPP (9) yields $\boldsymbol{w}_f = [4.04\ 2.54\ 1\ 0\ 0.5\ 1]^T$ $\tau_f = 2$ t.u.

By executing Algorithm 1 ($\Theta = 0.01$), $\boldsymbol{m}_f$ is reached by 122 discrete steps, which corresponds to 1.22 time unit (t.u.) Evolution of markings m_1, m_2 and m_3 and, control actions-controlled flows of transitions t_1 and t_2 are shown in Fig. 2 and 3, respectively.

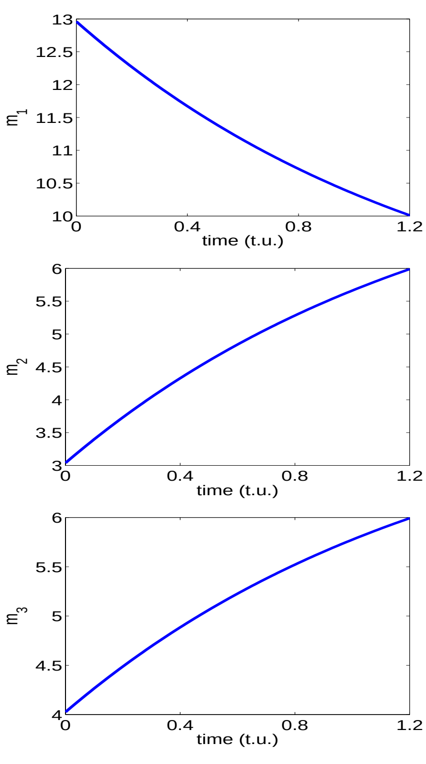

Fig. 2: Evolution of m_1 and m_2 for Example 1

IV. Final Marking Planning for Join Free contPNs

So far, we introduced an online control method for driving the system from a given initial state $\boldsymbol{m}_0$ to the

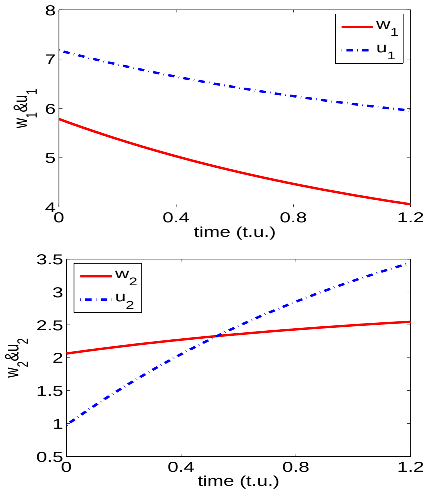

Fig. 3: Evolution of control actions and controlled flows of t_1 and t_2 for Example 1

specified target marking. In this section, we consider the control problems that final markings of some places are specified, while the others are not. The set of places whose final markings are specified precisely is denoted by $P_{sp} = \{p_i \mid m_{f_i}$ is specified$\}$ and the set of places whose final markings are not specified is denoted by $P_{un} = \{p_i \mid m_{f_i}$ is un-specified$\}$. In order to calculate the m_{f_i} $\forall p_i \in P_{un}$ by minimizing the time (or maximizing controlled flows $\boldsymbol{w}_0$ and $\boldsymbol{w}_f$), Algorithm 1 is used again with only one modification. Thus, as differ from the first case we propose to solve BLP in (19) instead of LPPs (8) and (9) with variables τ_0, τ_f, $\boldsymbol{s}_0$, $\boldsymbol{s}_f$, m_{f_i} $\forall p_i \in P_{un}$ where $\boldsymbol{s}_0 = \boldsymbol{w}_0 \cdot \tau_0$, $\boldsymbol{s}_f = \boldsymbol{w}_f \cdot \tau_f$:

$$\begin{array}{lll}
\min & \tau_0 + \tau_f & \\
\text{s.t.} & \boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{s}_0 & (a1) \\
& 0 \leq s_{0_j} \leq \lambda_j \cdot \Pi_{ji} \cdot m_{0_i} \cdot \tau_0 & \\
& \quad \forall\, j \in \{1,..,|T|\} \text{ where } i \text{ satisfies } \Pi_{ji} \neq 0 & (a_2) \\
& \boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{s}_f & (b1) \\
& 0 \leq s_{f_j} \leq \lambda_j \cdot \Pi_{ji} \cdot m_{f_i} \cdot \tau_f & \\
& \quad \forall\, j \in \{1,..,|T|\} \text{ where } i \text{ satisfies } \Pi_{ji} \neq 0 & (b_2) \\
& \boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma},\ \boldsymbol{\sigma} > \boldsymbol{0} & (c)
\end{array} \qquad (19)$$

The equations correspond to: (a1)&(b1) the equation of the straight line connecting $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$; (a2)&(b2) flow constraints at $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$, respectively; (c) reachability condition of $\boldsymbol{m}_f$. Note that, since the net we consider is consistent (19)(c) is equivalent to $\boldsymbol{B}_y^T \cdot \boldsymbol{m}_f = \boldsymbol{B}_y^T \cdot \boldsymbol{m}_0$, $\boldsymbol{m}_f > \boldsymbol{0}$ where $\boldsymbol{B}_y^T$ is basis of *P-semiflows* [13].

Example 2: Let us go back to Example 1 with the same initial marking $\boldsymbol{m}_0 = [13\ 3\ 4\ 4\ 5]^T$ and Θ. But in this case, final marking of some places (not all) are specified: $m_{f_1} = 10$, $m_{f_4} = 3$, $m_{f_5} = 2$, that is $P_{sp} = \{p_1, p_4,\ p_5\}$ and $P_{un} = \{p_2, p_3\}$. Our objective is to find the final markings of m_{f_2} and m_{f_3} by minimizing the time and drive the system to fulfilled final marking. For this example, BLP in (19) leads to:

$$
\begin{array}{lll}
\min & \tau_0 + \tau_f & \\
\text{s.t.} & 10 = 13 + s_{02} + s_{06} - s_{01} - s_{03} & \\
 & m_{f_2} = 3 + s_{01} + s_{02} & \\
 & m_{f_3} = 4 + s_{03} - s_{05} & (a1) \\
 & 6 = 4 + s_{03} - s_{05} & \\
 & 6 = 5 + s_{04} + s_{05} - 2 \cdot s_{06} & \\
 & & \\
 & 0 \le s_{01} \le 13 \cdot \tau_0 & \\
 & 0 \le s_{02} \le 3 \cdot \tau_0 & \\
 & 0 \le s_{03} \le 13 \cdot \tau_0 & (a2) \\
 & 0 \le s_{04} \le 4 \cdot \tau_0 & \\
 & 0 \le s_{05} \le 4 \cdot \tau_0 & \\
 & 0 \le s_{06} \le 2.5 \cdot \tau_0 & \\
 & & \\
 & 10 = 13 + s_{f_2} + s_{f_6} - s_{f_1} - s_{f_3} & \\
 & m_{f_2} = 3 + s_{f_1} + s_{f_2} & \\
 & m_{f_3} = 4 + s_{f_3} - s_{f_5} & (b1) \\
 & 6 = 4 + s_{f_3} - s_{f_5} & \\
 & 6 = 5 + s_{f_4} + s_{f_5} - 2 \cdot s_{f_6} & \\
 & & \\
 & 0 \le s_{f_1} \le 10 \cdot \tau_f & \\
 & 0 \le s_{f_2} \le m_{f_2} \cdot \tau_f & \\
 & 0 \le s_{f_3} \le 10 \cdot \tau_f & (b2) \\
 & 0 \le s_{f_4} \le m_{f_3} \cdot \tau_f & \\
 & 0 \le s_{f_5} \le 6 \cdot \tau_f & \\
 & 0 \le s_{f_6} \le 3 \cdot \tau_f & \\
 & & \\
 & 2 \cdot m_{f_2} + m_{f_3} = 18 & (c)
\end{array}
\qquad (20)
$$

By solving the BLP in (20), markings of unspecified places are obtained as $m_{f_2} = 6.5$ and $m_{f_3} = 5$, that is $\boldsymbol{m}_f = [10\ 6.5\ 5\ 3\ 2]^T$. By executing Algorithm 1, $\boldsymbol{m}_f$ is reached by 90 discrete steps (0.90 t.u.). Evolution of markings m_1, m_2 and m_3 and control actions-controlled flows of transitions t_1 and t_2 are shown in Fig. 4 and 5, respectively.

V. Conclusion

An online control method is developed for JF contPN. The method takes discrete time representation. In this method, in order to drive the system from its initial marking to target marking, corresponding control action is calculated and applied at each time step. Algorithm 1 which uses LPP is developed for this method.

In some control problems, target markings of some places are given while that of others are not specified. In that case, we propose to calculate unspecified target markings of places by solving a BLP with time minimization objective. Then Algorithm 1 is executed again with a simple modification.

An interesting point is to extend this work to more general structures, that is for other types of contPNs.

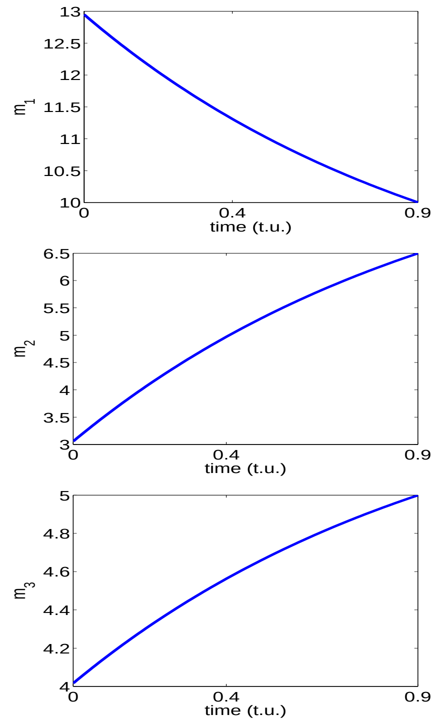

Fig. 4: Evolution of m_1 and m_2 for Example 2

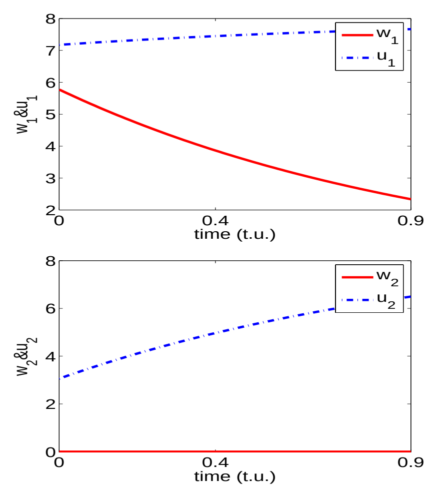

Fig. 5: Evolution of control actions and controlled flows of t_1 and t_2 for Example 2

References

[1] D. Mandrioli, A. Morzenti, M. Pezze, P. Pietro S., and S. Silva. A Petri net and logic approach to the specification and verification of real time systems. In *Formal Methods for Real Time Computing*. John Wiley & Sons Ltd., 1996.

[2] T. Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77:541–580, 1989.

[3] M. Silva and L. Recalde. Petri nets and integrality relaxations a view of continuous Petri nets. *IEEE Trans on Systems Man and Cybernetics*, 32(4):314–327, 2002.

[4] A. Aybar, A. İftar, and H. Apaydin-Özkan. Centralized and decentralized supervisory controller design to enforce boundedness, liveness, and reversibillity in petri nets. *International Journal of Control*, pages 537–553, 2005.

[5] A. Aybar and A. İftar. Decentralized structural controller design for large-scale discrete event systems modelled by petri nets. *Kybernetika*, 1(45):3–14, 2009.

[6] T. Nishi, K. Shimatani, and M. Inuiguchi. Lagrangian relaxation technique for solving scheduling problems by decomposition of timed petri nets. In *Proceedings of the 17th IFAC World Congress*, volume 17,

COEX, Korea, South, 2008.
[7] R. David and H.Alla. Continuous Petri nets. In *8th European Workshop on Application and Theory of Petri Nets*, Zaragoza, Spain, 1987.
[8] C. Mahulea, A. Giua, L. Recalde, C. Seatzu, and M. Silva. Optimal model predictive control of timed continuous Petri nets. *IEEE Transactions on Automatic Control*, 53(7):1731 – 1735, August 2008.
[9] J. Xu, L. Recalde, and M. Silva. Tracking control of join-free timed continuous Petri net systems under infinite servers semantics. *Discrete Event Dynamic Systems*, 18(2):263–288, 2008.
[10] J. Xu, L. Recalde, and M. Silva. Tracking control of timed continuous Petri net systems under infinite servers semantics. In *17th World Congress of IFAC*, pages 3192–3197, Seoul, Korea, 2008.
[11] C.R. Vázquez and M. Silva. Piecewise-linear constrained control for timed continuous petri nets. In *IEEE Conference on Decision and Control*, Shanghai, China, 12/2009 2009.
[12] Hanife Apaydin-Özkan, Jorge Júlvez, Cristian Mahulea, and Manuel Silva. Approaching minimum time control of timed continuous petri nets. *Nonlinear Analysis: Hybrid Systems*, 5(2):136 – 148, 2011. Special Issue related to IFAC Conference on Analysis and Design of Hybrid Systems (ADHS'09) - IFAC ADHS'09.
[13] 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.
[14] H.Alla and R.David. Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers*, 8:159–188, 1998.
[15] 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.
[16] J. Júlvez, L. Recalde, and M. Silva. On reachability in autonomous continuous Petri net systems. In *24th International Conference on Application and Theory of Petri Nets*, pages 221–240, Eindhoven, Netherlands, June 2003. Springer.