Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Formal Analysis of Timed Continuous Petri Nets

Marius Kloetzer, Cristian Mahulea, Calin Belta, Laura Recalde and Manuel Silva

Abstract— **In this paper, we develop an automated framework for formal verification of timed continuous Petri nets (*contPN*). Specifically, we consider two problems: (1) given an initial set of markings, construct a set of unreachable markings, i.e., such that all trajectories starting in the initial set avoid the latter one; (2) given a Linear Temporal Logic (LTL) formula over a set of linear predicates in the state, construct a set of initial states such that all trajectories originating there satisfy the specification. The starting point for our approach is the observation that a *ContPN* system can be written as a Piecewise Affine (PWA) system with a polyhedral partition. We propose an iterative method for analysis of PWA systems from specifications given as LTL formulas over linear predicates. The computation consists of polyhedral operations and searches on graphs only. We present two illustrative numerical examples.**

I. INTRODUCTION

As genuine paradigms for modeling discrete event dynamic systems, basic (fully non deterministic) Petri nets suffer from state explosion problems; even more, several undecidability results appear, if they are constrained with time extensions. Additionally, computational aspects usually become drastically more complex as the initial marking (set of resources and servers) increases. In this context, fluidification of Petri nets [1], [2], [3] is a promising relaxation, particularly useful (errors tend to be smaller, and computational savings to be bigger) when "significant" initial markings are considered. For example, fluidification transforms into polynomial time complexity the computation of the optimal initial marking for several steady state problems of practical interest [2]. Nevertheless, this relaxation does not necessarily mean that even basic properties of continuous but timed net models must become decidable [4]. Moreover, it is interesting to explore the use of formal analysis techniques using temporal logic on the polytope containing the markings of the relaxed net model, which is the topic of this paper.

Technically, the approach presented in this paper is based on the PWA representation of the dynamics of a deterministically timed continuous Petri nets, under so called infinite server semantics (a direct transposition of the corresponding firing flow definition of discrete Petri nets, under markovian interpretation [5]). The use of PWA powerful analysis tools is an immediate and interesting consequence.

This work was partially supported by the projects CICYT - FEDER DPI2003-06376 and DPI2006-15390 at the University of Zaragoza and by NSF CNS-0410514 at Boston University.

M. Kloetzer and C. Belta are with the Center for Information and Systems Engineering, Boston University, Brookline, MA 02446, USA, {kmarius,cbelta}@bu.edu

C. Mahulea, L. Recalde and M. Silva are with the Aragón Institute for Engineering Research (I3A), University of Zaragoza, 50018 Zaragoza, Spain, {cmahulea,lrecalde,silva}@unizar.es

From a different perspective, fluidification of discrete Petri nets can be viewed as a relaxation of integer programming problems into convex geometry-linear programming ones [6], allowing for structural analysis techniques, where the initial marking is abstracted, or managed in a parametric way. In this sense, the approach in this work tries to abstract from a precise value for the initial marking, or to compute regions of initial markings for which some LTL formula is true for the infinite set of trajectories that are generated.

The contribution of the paper is twofold. First, as already stated, it provides a fully automatic framework for formal analysis of timed continuous Petri nets. Second, as part of this framework, we developed a general-use tool for formal analysis of PWA systems with continuous vector fields from temporal logic formulas over linear predicates. This relates to [7], where a richer class of hybrid affine systems is analyzed against reachability properties only.

Temporal logic analysis problems for PWA systems are also studied in [8] (in continuous time) and [9] (in discrete time). However, in these works, the refinement is based on an (approximate) implementation of the bisimulation algorithm, and on the computation of the Pre image of sets through the vector fields of the system. In this paper, the iterative refinement is achieved through simple cuts, and resembles our previous work [10] for multi-affine systems and rectangular sets.

After some preliminaries concerning Petri nets, transition systems and temporal logic (Section II), the addressed problems are formulated and translated to PWA formulations (Section III). Formal analysis of PWA systems is presented in Section IV, while Sections V and VI provide some simulation results and concluding remarks.

II. PRELIMINARIES

A. Timed Continuous Petri Nets

***Definition** 2.1:* [Continuous Petri Net System] A Continuous Petri Net (*ContPN*) 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* and $\boldsymbol{m}_0 \in \mathbb{R}_{\geq 0}^{|P|}$ is the *initial marking*. P is the set of places, T is the set of transitions, and $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{N}^{|P| \times |T|}$ are the pre and post incidence matrices, respectively.

Let p_i, $i = 1, \ldots, |P|$ and t_j, $i = 1, \ldots, |T|$ denote the places and transitions. 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 token load denoted by $m_i \in \mathbb{R}_{\geq 0}$. The vector of all token loads is called marking, and is denoted by $\boldsymbol{m} \in \mathbb{R}_{\geq 0}^{|P|}$. The *preset* and *postset* of a place or transition $x \in P \cup T$ are

denoted by ${}^\bullet x$ and $x^\bullet$, and represent the input and output transitions and places of x, respectively. Precisely, if $t_i \in T$, ${}^\bullet t_i = \{p_j \in P | Pre_{ji} > 0\}$ and $t_i{}^\bullet = \{p_j \in P | Post_{ji} > 0\}$. Otherwise, if $p_i \in P$, ${}^\bullet p_i = \{t_j \in T | Post_{ij} > 0\}$ and $p_i{}^\bullet = \{t_j \in T | Pre_{ij} > 0\}$.

It is important to note that the marking of a *ContPN* can take real positive values, while in discrete Petri Nets (PN) only natural values are possible. In fact, this is the only difference between a continuous and a discrete PN.

A transition $t_j \in T$ is *enabled* at $\boldsymbol{m}$ iff $\forall p_i \in {}^\bullet t_j$, $m_i > 0$. Its enabling degree is

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

which represents the maximum amount in which t_j can fire. An enabled transition t_j can fire in any real amount $0 < \alpha < enab(t_j, \boldsymbol{m})$ leading to a new marking $\boldsymbol{m'} = \boldsymbol{m} + \alpha \boldsymbol{C}_{\cdot j}$, 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, a *state (or fundamental) equation* can be written: $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, where $\boldsymbol{\sigma} \in \mathbb{R}_{\geq 0}^{|T|}$ is the firing count vector, i.e., σ_j is the cumulative amount of firing of t_j in the sequence σ. The set of all reachable markings from $\boldsymbol{m}_0$ is called the *reachability space* and it is denoted by $\mathcal{R}(\mathcal{N}, \boldsymbol{m}_0)$, or simply by $\mathcal{R}$ when there is no confusion on $\mathcal{N}$ and $\boldsymbol{m}_0$. In the case of a *ContPN* system, $\mathcal{R}$ is a convex set [2].

A *ContPN* is *bounded* when every place is bounded (for all $p_i \in P, \exists b_i \in \mathbb{R}_{\geq 0}$ with $m_i \leq b_i$ at every reachable marking $\boldsymbol{m}$). Right and left non negative annullers of $\boldsymbol{C}$ are called T- and P-semiflows, respectively. If non negativity is not required, the annullers are called T- and P-flows.

If a timed interpretation is included in the model, the fundamental equation depends on time: $\boldsymbol{m}(\tau) = \boldsymbol{m_0} + \boldsymbol{C} \cdot \boldsymbol{\sigma}(\tau)$, which, through time differentiation, becomes $\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \dot{\boldsymbol{\sigma}}(\tau)$. The derivative of the firing sequence $\boldsymbol{f}(\tau) = \dot{\boldsymbol{\sigma}}(\tau)$ is called the *(firing) flow*, and leads to the following equation for the dynamics of the *ContPN*:

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

This paper deals with *infinite server semantics*, which was shown to provide a good approximation of the underlying discrete net for a broad class of systems [11]. Under this semantics, the flow of transition t_j is given by:

$$f_j(\tau) = \lambda_j \; enab(t_j, \boldsymbol{m}(\tau)), \tag{3}$$

where $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{|T|}$ associates a constant value $\lambda_j > 0$, representing its firing rate, to each transition t_j, and the enabling function is given by (1). From (2), (3), and (1), it can be easily seen that a *ContPN* system with infinite server semantics is a piecewise linear system with polyhedral regions and everywhere continuous vector field. In other words, the dynamics of the markings are given by

$$\dot{\boldsymbol{m}}(\tau) = \boldsymbol{A}_i \boldsymbol{m}(\tau), \; \boldsymbol{m} \in \mathcal{R}_i, \; i \in I, \tag{4}$$

where $\boldsymbol{A}_i \in \mathbb{R}^{|P| \times |P|}$, $\mathcal{R}_i$ is a polyhedral set, and I is a set of labels for the modes of the piecewise linear system (see [12] for more details).

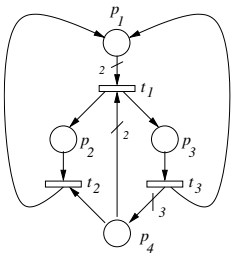


Fig. 1: A timed continuous Petri Net *ContPN*.

The number of regions $\mathcal{R}_i$ of a *ContPN* system is upper bounded by $\prod_{t_i \in T} |{}^\bullet t_i|$ and in the case of a bounded net system they are closed polytopes. For a given initial marking, some places can be implicit [6] (given a *ContPN* system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$, $p_j \in {}^\bullet t_i$ is implicit iff $\nexists \boldsymbol{m} \in \mathcal{R}(\mathcal{N}, \boldsymbol{m}_0)$ such that $\frac{m_j}{Pre_{ji}} < \frac{m_k}{Pre_{ki}} \; \forall p_k \in {}^\bullet t_i \setminus \{p_j\}$). For example, in the *ContPN* in Fig. 1 with $\boldsymbol{m}_0 = [15, 3, 1, 0]^T$, p_2 is an implicit place. Therefore, the region $\mathcal{R}_3 = \{\frac{m_1}{2} \leq \frac{m_4}{2}, m_2 \leq m_4\}$ is included in $\mathcal{R}_1 = \{\frac{m_1}{2} \leq \frac{m_4}{2}, m_4 \leq m_2\}$ since $m_2 \leq m_4$ is satisfied only as equality. In fact, $\mathcal{R}_3$ is a frontier of $\mathcal{R}_1$. Also, $\mathcal{R}_4 = \{\frac{m_4}{2} \leq \frac{m_1}{2}, m_2 \leq m_4\}$ is included in $\mathcal{R}_2 = \{\frac{m_4}{2} \leq \frac{m_1}{2}, m_4 \leq m_2\}$ for the same reason. In our approach we consider only the regions that are full-dimensional polytopes in $\mathbb{R}^{rank(\boldsymbol{C})}$. Note that this is not a limitation since at the common border of two regions, the corresponding linear systems provide the same vector field according to (3) and (1).

B. Transition systems and temporal logic

***Definition** 2.2:* [Transition system] A transition system is a tuple $\mathcal{T} = (Q, Q_0, \rightarrow, \Pi, \models)$, where Q is a (possibly infinite) set of states, $Q_0 \subseteq Q$ is a set of initial states, $\rightarrow \subseteq Q \times Q$ is a transition relation, Π is a finite set of atomic propositions, and $\models \subseteq Q \times \Pi$ is a satisfaction relation.

For an arbitrary proposition $\pi \in \Pi$, we define $[\![\pi]\!] = \{q \in Q | q \models \pi\}$ as the set of all states satisfying it. Conversely, for an arbitrary state $q \in Q$, let $\Pi_q = \{\pi \in \Pi \,|\, q \models \pi\}$, $\Pi_q \in 2^\Pi$, denote the set of all atomic propositions satisfied at q. An initialized *trajectory* or *run* of $\mathcal{T}$ starting from $q \in Q_0$ is an infinite sequence $r = r(1)r(2)r(3)\ldots$ with the property that $r(1) = q$, $r(i) \in Q$, and $(r(i), r(i+1)) \in \rightarrow$, for all $i \geq 1$. A trajectory $r = r(1)r(2)r(3)\ldots$ generates a *word* $w = w(1)w(2)w(3)\ldots$, where $w(i) = \Pi_{r(i)}$. The set of all generated words is called the *language* of $\mathcal{T}$, and is denoted by $L(\mathcal{T})$.

An equivalence relation $\sim \subseteq Q \times Q$ over the state space of $\mathcal{T}$ is *proposition preserving* if for all $q_1, q_2 \in Q$ and all $\pi \in \Pi$, if $q_1 \sim q_2$ and $q_1 \models \pi$, then $q_2 \models \pi$. A proposition preserving equivalence relation naturally induces a *quotient transition system* $\mathcal{T}/_\sim = (Q/_\sim, Q_0/_\sim, \rightarrow_\sim, \Pi, \models_\sim)$. $Q/_\sim$ is the quotient space (the set of all equivalence classes), and the set of initial states is $Q_0/_\sim = \{P \in Q/_\sim \,|\, Q_0 \cap h_\sim(P) \neq \emptyset\}$, where $h_\sim : Q/_\sim \rightarrow 2^Q$ is the concretization map corresponding to $\sim$. The transition relation $\rightarrow_\sim$ is defined as follows: for $P_1, P_2 \in Q/_\sim$, $P_1 \rightarrow_\sim P_2$ if and only if there exist $q_1 \in h(P_1)$ and $q_2 \in h(P_2)$ such that $q_1 \rightarrow q_2$. The

satisfaction relation is defined as follows: for $P \in Q/_\sim$, we have $P \models_\sim \pi$ if and only if there exist $q \in h(P)$ such that $q \models \pi$. It is easy to see that

$$L(\mathcal{T}) \subseteq L(\mathcal{T}/_\sim), \tag{5}$$

The quotient transition system $\mathcal{T}/_\sim$ is said to *simulate* the original system $\mathcal{T}$, which is written as $\mathcal{T}/_\sim \geq \mathcal{T}$.

In this work we consider system specifications given as formulas of Linear Temporal Logic (LTL) [13]. A formal definition for the syntax and semantics of LTL formulas is beyond the scope of this paper. The LTL formulas are recursively defined over a set of atomic propositions Π, by using the standard boolean operators and a set of temporal operators, which include $\mathcal{U}$ ("until"), $\Box$ ("always"), $\Diamond$ ("eventually"). LTL formulas are interpreted over infinite words over 2^Π, such as those generated by the transition system $\mathcal{T}$ from Definition 2.2. If ϕ_1 and ϕ_2 are two LTL formulas over Π and w is a word produced by $\mathcal{T}$, then formula $\phi_1 \mathcal{U} \phi_2$ means that (over the word w) ϕ_2 will eventually become true, and ϕ_1 is true until this happens. Formula $\Diamond\phi_1$ means that ϕ_1 becomes eventually true, whereas $\Box\phi_1$ indicates that ϕ_1 is true at all positions of w. More expressiveness can be achieved by combining the mentioned operators.

***Remark** 2.3:* Because of the particular semantics of temporal logic formulas over continuous trajectories (see [14] for a detailed discussion), here we restrict our attention to $\mathrm{LTL}_{-\mathrm{X}}$, a subclass of LTL that lacks the "next" temporal operator. A careful examination of the LTL and $\mathrm{LTL}_{-\mathrm{X}}$ semantics shows that the increased expressivity of LTL is manifested only over words with a finite number of successive repetitions of a symbol. This property is also known as closure under stuttering of $\mathrm{LTL}_{-\mathrm{X}}$ [15]. In the rest of this work, whenever we refer to a generated word, we assume that all finite successive repetitions of the same symbol (subset of Π) is replaced by just one occurrence. The closure under stuttering of $\mathrm{LTL}_{-\mathrm{X}}$ guarantees that the satisfaction of a $\mathrm{LTL}_{-\mathrm{X}}$ formula is not influenced by such replacements. For simplicity of notation, we use LTL instead of $\mathrm{LTL}_{-\mathrm{X}}$ in the remainder of the paper.

Given a transition system $\mathcal{T}$ and an LTL formula ϕ over its set of propositions, checking whether $L(\mathcal{T})$ satisfies ϕ is called model checking. For finite transition systems, there exist off-the-shelf tools for model checking [16]. Note that if a proposition-preserving quotient $\mathcal{T}/_\sim$ satisfies ϕ, then by the language inclusion (5), the initial transition system $\mathcal{T}$ also satisfies the formula.

III. PROBLEM FORMULATION AND APPROACH

Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a *ContPN* system and Π be a set of strict linear inequalities over its marking $\boldsymbol{m}$, which will be simply called predicates. We assume that the set Π includes all the affine functions in $\boldsymbol{m}$ necessary to define the full-dimensional regions $\mathcal{R}_i$.

***Problem** 3.1 (Construction of safe sets):* Given a set of initial states defined as the conjunction of predicates from a set $\Pi_0 \subseteq \Pi$, find a subset of the state space that cannot be reached by trajectories of *ContPN* originating in the initial set.

***Problem** 3.2 (Verification):* Given an LTL formula over Π, find a set of initial states of *ContPN* from where all possible trajectories satisfy the formula.

To fully specify Problems 3.1 and 3.2, we need to define the semantics of an LTL formula over a continuous trajectory. A formal definition is given in Section IV through an embedding into a transition system. However, an informal and intuitive definition can be given as follows: an evolving trajectory produces the set of predicates from Π that are true at the current state, with no finite consecutive repetitions of the set of predicates, and with infinitely many repetitions of the set of predicates satisfied by a region that is an invariant for the trajectory. Note that this is consistent with our choice of LTL without the "next" operator (see Remark 2.3). For example, in Fig. 2, if the regions $\mathcal{R}_i$ satisfy the sets of predicates $\Pi_i \subseteq \Pi$, $i = 1, \ldots, 4$, respectively, then the shown trajectory, starting from $\boldsymbol{m}_0$ and converging to $\boldsymbol{m}_f$, generates the word $\Pi_1\Pi_2\Pi_4\Pi_3\Pi_3 \ldots$.

Our approach to solving Problems 3.1 and 3.2 consists of two main steps. First, we compute a set of linearly independent P-flows of *ContPN* and construct a reduced representation of the *ContPN* in the form of a piecewise affine system (PWA). Second, we perform formal analysis of the corresponding PWA system based on discrete abstractions (finite quotients) and refinement, and by employing convexity properties of affine systems in full-dimensional polytopes [17], [18], [14], as shown in Section IV.

The token conservation laws (P-flows) introduce a number of $|P| - rank(\boldsymbol{C})$ dependent variables [19]. These variables can be removed making some simple algebraic computations and a reduced system is obtained, but now the behavior is piecewise affine. For simplicity, we abuse the notation and denote the obtained PWA system by

$$\dot{\boldsymbol{m}}(\tau) = \boldsymbol{A}_i \boldsymbol{m}(\tau) + \boldsymbol{b}_i,\ \boldsymbol{m} \in \mathcal{R}_i,\ i \in I, \tag{6}$$

with the implicit understanding that the state $\boldsymbol{m}$ has already been reduced and $\boldsymbol{A}_i$ are the corresponding new system matrices. The regions $\mathcal{R}_i$ and the set I are the same as in (4), with the observation that $\mathcal{R}_i$ are now expressed using a smaller number of variables. The linear inequalities from the set of specification predicates Π are also transformed accordingly, while the predicate symbols remain the same. It is also easy to see that, as in the piecewise linear representation, the vector field of (6) is continuous everywhere.

The trajectories of the PWA system (6) produce words according to the informal definition above. In the rest of the paper, when we refer to Problems 3.1 and 3.2, we assume that they are formulated for the PWA representation (6) of the *ContPN* system.

***Example** 3.3:* The net in Fig. 1 has two token conservation laws (P-semiflows), thus two variables are redundant. If m_1 and m_4 are chosen as free variables, then a planar PWA representation of the form (6) can be constructed. The reachability space in the reduced (m_1, m_4) - plane is

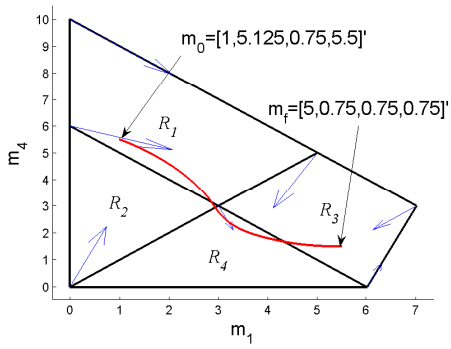


Fig. 2: Reachability space of the *ContPN* in fig. 1 with $\boldsymbol{m}_0 = [1, 5.125, 0.75, 5.5]^T$ and $\boldsymbol{\lambda} = \mathbf{1}$.

sketched in Fig. 2. The dynamics corresponding to region $\mathcal{R}_1$ are given by:

$$\begin{bmatrix} \dot{m}_1 \\ \dot{m}_4 \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} m_1(\tau) \\ m_4(\tau) \end{bmatrix} + \begin{bmatrix} 7 \\ 3 \end{bmatrix} \quad (7)$$

IV. FORMAL ANALYSIS OF PWA SYSTEMS

Assume there are M feasible sets of the form $\bigwedge_{i=1}^{|\Pi|}((-1)^{j_i}\pi_i)$, where $j_1, \ldots j_{|\Pi|} \in \{0, 1\}$. Since the affine functions necessary to define the regions $\mathcal{R}_k$ are among π_i, $i = 1, \ldots, |\Pi|$, each of these sets is a full dimensional polytope included in the reachability set of the PWA system, and corresponds to a feasible combination of predicates from Π inside each region $\mathcal{R}_k$. We denote these sets by $\mathcal{P}_1$,$\mathcal{P}_2$,...,$\mathcal{P}_M$.

***Definition** 4.1:* For the PWA system (6) and the set of predicates Π, the (infinite) embedding transition system is defined as

$$\mathcal{T}_{emb} = \{Q_{emb}, Q_{emb_0}, \rightarrow_{emb}, \Pi_{emb}, \vDash_{emb}\}, \quad (8)$$

where $Q_{emb} = \bigcup_{i=1}^{M} \mathcal{P}_i$, $Q_{emb_0} = Q_{emb}$, and $\Pi_{emb} = \Pi$. The satisfaction relation is obviously defined as $\boldsymbol{m} \vDash_{emb} \pi_i$ if and only if $\boldsymbol{m}$ verifies the strict linear equality π_i. The transition relation is defined according to the following two rules: (1) $(\boldsymbol{m'}, \boldsymbol{m''}) \in \rightarrow_{emb}$ with $\boldsymbol{m'} \in \mathcal{P}_i$, $\boldsymbol{m''} \in \mathcal{P}_j$, $\mathcal{P}_i \neq \mathcal{P}_j$ if and only if the polytopes $\mathcal{P}_i$ and $\mathcal{P}_j$ are adjacent[1] and there exist a trajectory $\boldsymbol{m}(\tau)|_{[0,T]}$ of (6) ($0 < T < \infty$) such that $\boldsymbol{m}(0) = \boldsymbol{m'}$, $\boldsymbol{m}(T) = \boldsymbol{m''}$, and $\boldsymbol{m}(\tau)|_{[0,T]}$ is included in the closure of $\mathcal{P}_i \bigcup \mathcal{P}_j$; (2) $(\boldsymbol{m'}, \boldsymbol{m''}) \in \rightarrow_{emb}$ with $\boldsymbol{m'}, \boldsymbol{m''} \in \mathcal{P}_i$ if and only if there exist a trajectory $\boldsymbol{m}(\tau)|_{[0,\infty)}$ of (6) such that $\boldsymbol{m'} = \boldsymbol{m}(0)$ and $\boldsymbol{m''} = \lim_{\tau \to \infty} \boldsymbol{m}(\tau)$.

Note that the trajectories of $\mathcal{T}_{emb}$ satisfy the informal definition from Section III. Formally, we have:

***Definition** 4.2:* The set of all words produced by trajectories of the PWA system (6) representing the *ContPN* system, is the language $L(\mathcal{T}_{emb})$ of the transition system (8).

***Remark** 4.3:* For technical reasons, we limit the specifications in the set of predicates Π to strict linear inequalities. However, this assumption does not seem restrictive from an application point of view. If the predicates in Π model sensor information, it is unrealistic to check for the attainment of a specific value due to sensor noise. Moreover, if a specific value is of interest, it can be included in the interior of a polytope given by other predicates. In addition, in the definition of $\mathcal{T}_{emb}$, we ignore the states of *contPN* that lie on the hyperplanes obtained by setting to zero the linear inequalities from π_i, $i = 1, \ldots, |\Pi|$. This might lead to problems in the case when there are trajectories originating at states of $\mathcal{T}_{emb}$ and "disappearing" inside such hyperplanes. However, it can be easily shown that such situations cannot occur if the vector field is continuous everywhere, which is the case here.

The embedding transition system (8) has infinitely many states and cannot be model checked. To provide (conservative) solutions to Problems 3.1 and 3.2, we propose an iterative procedure that produces a finite quotient and then refines it if necessary. At each step, the language of the obtained quotient includes the language of $\mathcal{T}_{emb}$.

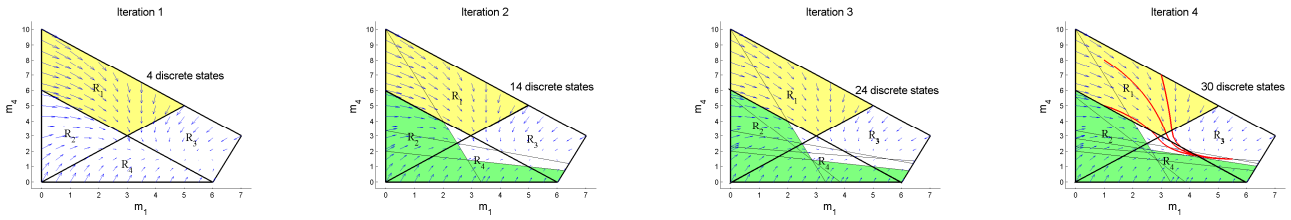
A. Construction and analysis of the quotients

Let $\sim$ be a polytopal proposition-preserving equivalence relation over Q_{emb} that does not violate the polytopes $\mathcal{P}_i$, $i = 1, \ldots, M$. In other words, each equivalence class in $Q_{emb}/_\sim$ is a polytope included in exactly one of $\mathcal{P}_i$, $i = 1, \ldots, M$. According to Definition 4.1, to compute the transitions of $\mathcal{T}_{emb}/_\sim$, we need to solve the following two problems: (i) for all pairs of equivalence classes corresponding to adjacent polytopes, decide if there is a trajectory of $\mathcal{T}_{emb}$ penetrating from one to another, and (ii) for all equivalence classes, decide if there exist a trajectory of $\mathcal{T}_{emb}$ for which the corresponding polytope is an invariant.

For both problems (i) and (ii) above, we propose to use the computational framework developed in [18]. Due to space limitations, we omit the details and outline the main ideas only. Specifically, in [18], it is shown that an affine system has a trajectory contained in a full dimensional open polytope for all times if and only if the affine system has an equilibrium inside the polytope. Therefore, solving problem (ii) in a polytopal equivalence class reduces to checking the non-emptiness of the polyhedral set given by the equations of the polytope plus the equation setting the corresponding vector field to zero. In addition, in [18], it is shown that, given two adjacent polytopes, there exists a trajectory penetrating from one to another in finite time if and only if there exists a vertex on the common facet at which the projection of the vector field on the outer normal of the facet pointing from the first to the latter is strictly positive. Recall that the vector field of our system is continuous everywhere, so the vector fields of two affine systems on adjacent polytopes agree on the common facet. In conclusion, solving both problems (i) and (ii) reduces to checking non-emptiness of polyhedral sets, for which there exist several powerful tools [20].

Having a finite quotient $\mathcal{T}_{emb}/_\sim$, we can provide a (conservative) solution to Problem 3.1 as follows. First, we define the set of initial states $Q_{emb_0}/_\sim$ as the set of states

[1]Throughout the paper, we call two full dimensional polytopes in $\mathbb{R}^N$ adjacent if they share a facet that is a full dimensional polytope in $\mathbb{R}^{N-1}$.



(a) (b) (c) (d)

Fig. 3: Iterative construction of a safe set for the initial region shown in yellow. The safe set obtained at each iteration is shown in green.

of $\mathcal{T}_{emb}/_{\sim}$ that satisfy the predicates from Π_0. Then, by using a simple search on a graph, we find all states Q_{nr} of $\mathcal{T}_{emb}/_{\sim}$ that are not reachable from $Q_{emb_0}/_{\sim}$. Enabled by the language inclusion property (5), a solution to Problem 3.1 can be presented in the form $\{\boldsymbol{m} \in h_{\sim}(q) \,|\, q \in Q_{nr}\}$, where $h_{\sim}$ is the concretization map defined in Section II-B.

The LTL verification Problem 3.2 can be solved by model checking $\mathcal{T}_{emb}/_{\sim}$ from each initial state using an off-the-shelf model checker. If the formula is satisfied at a state q of $\mathcal{T}_{emb}/_{\sim}$, then, by the language inclusion property (5), all trajectories of $\mathcal{T}_{emb}$ (and of *ContPN*) starting at $h_{\sim}(q)$ satisfy the formula. If we denote by Q_s the set of all initial states of $\mathcal{T}_{emb}/_{\sim}$ from which the formula is satisfied, then a set of initial states of $\mathcal{T}_{emb}$ (and of *ContPN*) from which the formula is satisfied is given by $\{\boldsymbol{m} \in h_{\sim}(q) \,|\, q \in Q_s\}$. In our implementation, we used our own LTL planning tool developed in [14] and further improved in [21]. This is computationally more attractive, because our algorithm reuses some computations from the previously considered initial state, instead of completely reiterating a model checker for each new initial state (for details, see [21]).

B. Iterative analysis and refinement

We first construct and analyze the "roughest" quotient $\mathcal{T}_{emb}/_{\sim}$, which corresponds to partitioning with respect to predicates from the initial set Π, and to the equivalence relation defined by $\boldsymbol{m} \sim \boldsymbol{m'}$ if and only if there exist $\mathcal{P}_i$, $i = 1, \ldots, M$, such that $\boldsymbol{m}, \boldsymbol{m'} \in \mathcal{P}_i$. If the safe set is not large enough (or empty) in Problem 3.1, or if the set of initial states in not large enough (or empty) in Problem 3.2, then we construct "finer" quotients.

***Example** 4.4:* For the *ContPN* from Fig. 1 with $\boldsymbol{m}_0 = [1, 5.125, 0.75, 5.5]^T$ and $\boldsymbol{\lambda} = \mathbf{1}$, if the set Π contains only the linear predicates necessary to define the regions $\mathcal{R}_i$, $i = 1, 2, 3, 4$, then the first quotient is shown in Fig. 4. If we are interested in constructing a safe set (Problem 3.1), then it is easy to see that this set is empty. However, this set becomes non-empty through refinement, as shown below.

We construct finer quotients by adding to the current set Π some new predicates (from a set $\mathcal{H}$), and then recomputing the new feasible polytopes $\mathcal{P}_i$, as explained at the beginning of Section IV. Let us denote by $\mathcal{T}_{emb}/_{\sim}^{\mathcal{H}}$ the quotient obtained as in section IV-A, but corresponding to the set of predicates $\Pi \cup \mathcal{H}$ instead of Π (for simplicity and since no confusion is possible, we use the same notation $\sim$ for the

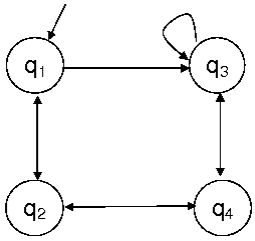


Fig. 4: The first quotient of the PWA system from Fig. 2.

polytopal proposition-preserving equivalence relation, even if it refers to a new partition). It is immediate to observe that $\mathcal{T}_{emb}/_{\sim} \geq \mathcal{T}_{emb}/_{\sim}^{\mathcal{H}}$, simply because the new partition[2] is a subpartition of the one corresponding to $\mathcal{T}_{emb}/_{\sim}$. Therefore, $L(\mathcal{T}_{emb}/_{\sim}) \supseteq L(\mathcal{T}_{emb}/_{\sim}^{\mathcal{H}}) \supseteq L(\mathcal{T}_{emb})$, which means that by using $\mathcal{T}_{emb}/_{\sim}^{\mathcal{H}}$ instead of $\mathcal{T}_{emb}/_{\sim}$ we can obtain less conservative solutions for Problems 3.1 and 3.2.

We start with $\mathcal{H} = \emptyset$, and for each pair of states $q_i, q_j \in Q_{emb}/_{\sim}$, $i < j$, such that $(q_i, q_j) \in \rightarrow_{emb} /_{\sim}$ and $(q_j, q_i) \in \rightarrow_{emb} /_{\sim}$, a new predicate α is added to $\mathcal{H}$, with α denoting the halfspace that separates on the common facet of $h_{\sim}(q_i)$ and $h_{\sim}(q_j)$ the points where the vector field projection on the outer normal of the common facet has positive and negative values, respectively. Assumption $i < j$ guarantees that we do not create two propositions for the same pair of states of $\mathcal{T}_{emb}/_{\sim}$. Results from [18] guarantee that such a separation is possible by a single linear predicate. For avoiding some new notations, we do not include the explicit equation of α, and we just mention that its computation requires only matrix multiplications. Our method of adding transitions between states of the discrete quotients implies that α can help in increasing the difference between $L(\mathcal{T}_{emb}/_{\sim})$ and $L(\mathcal{T}_{emb}/_{\sim}^{\mathcal{H}})$, as explained next.

Assume that $h_{\sim}(q_i)$ and $h_{\sim}(q_j)$ are each split by α in two subpolytopes, labelled in $\mathcal{T}_{emb}/_{\sim}^{\mathcal{H}}$ by q_i', q_i'', and q_j', q_j'', respectively. Note that q_i' and q_i'' are adjacent, and each of them is adjacent with only one of q_j', q_j'' (not with both), and vice-versa. Assume that q_i' is adjacent with q_j' and q_i'' is adjacent with q_j''. Then, the above mentioned sign separation provided by α, and the way of adding transitions from

[2]The regions induced by the proposition-preserving equivalence relation at each step do not really produce a partition of the state space. Because we consider only strict inequalities, we "lose" points at each step.

section IV-A, guarantee that in $\mathcal{T}_{emb}/_{\sim}^{\mathcal{H}}$ there exist either transitions (q_i', q_j') and (q_j'', q_i''), or transitions (q_j', q_i') and (q_i'', q_j''). Therefore, we hope that $\mathcal{T}_{emb}/_{\sim}^{\mathcal{H}}$ is less conservative than $\mathcal{T}_{emb}/_{\sim}$ (this fact cannot be guaranteed before testing transitions between q_i' and q_i'', q_j' and q_j'', respectively, and these transitions are not resulting from properties of α, but from tests as in section IV-A).

Note that there are infinitely many choices of predicates α yielding the same separation of the common facet of $h_{\sim}(q_i)$ and $h_{\sim}(q_j)$. Alternatively, one can focus on different splitting methods (instead of linear predicates), as long as the same sign separation is enforced. The motivation for our choice of cutting is three-fold. First, α is very easy to compute, and second, when splitting with some additional linear predicates we use the same algorithms as before, but with a larger input set Π. Third, we have the guarantee that the adjacent polytopes from partition exactly share facets (as needed for adding transitions in discrete quotients). The drawback is that α will not split only $h_{\sim}(q_i)$ and $h_{\sim}(q_j)$, but also other polytopes from the partition corresponding to $\mathcal{T}_{emb}/_{\sim}$, and thus the number of states of $\mathcal{T}_{emb}/_{\sim}^{\mathcal{H}}$ can increase significantly. Another way of cutting could involve a triangulation of $h_{\sim}(q_i)$ and $h_{\sim}(q_j)$ that preserves (contains as edge) the sign separating set we want. However, there are no algorithms for performing such a constrained triangulation in space dimensions higher than 2.

Even if the solutions to Problems 3.1 and 3.2 at a given step are not satisfactory, there are two situations when we do not perform refinement: either no more predicates are found, or a certain imposed complexity limit is reached (*e.g.*, a maximum number of states in the discrete quotient is reached). We note that, even if refinement in the current step does not produce a better solution to one of our problems, the refinement in the next step might yield an improvement, as it can be seen in the next example.

***Example** 4.5:* Consider the *ContPN* system in Fig. 1 with $\boldsymbol{m}_0 = [1, 5.125, 0.75, 5.5]^T$, $\boldsymbol{\lambda} = \mathbf{1}$ and the problem of constructing a safe set (Problem 3.1) for the initial region $\mathcal{R}_1$. It has been seen in Ex. 4.4 that at the first iteration, no safety regions are obtained (Fig. 3a). Through refinement, at the second step the transition system will contain 14 discrete states and a safety region depicted in Fig. 3b. At the next iteration, the number of discrete states of the transition system grows to 24, but the safety region is exactly the same as in previous step (Fig. 3c). Refining more, a number of 30 discrete states is obtained and the safety region is increased a little (Fig. 3d). Since no other cutting is possible the iteration is finished.

V. CONCLUSION

The focus of this paper was on developing automated frameworks for formal analysis of timed continuous Petri nets. We addressed two important problems, namely a safety problem, and an LTL verification problem. The solutions for both these problems started by reducing the initial *ContPN* to an equivalent piecewise affine system. Then, a finite (and conservative) abstraction of this system was constructed by using computationally attractive results that mainly involve polyhedral operations. Intermediate solutions for the initial problems were obtained by using the discrete abstraction and standard tools as searches on graphs and model checking algorithms. Finally, a refinement procedure was developed, allowing us to iteratively reduce the modelling conservativeness and improve the solutions to our initial problems.

REFERENCES

[1] R. David and H. Alla, *Discrete, Continuous and Hybrid Petri Nets*. Springer-Verlag, 2005.

[2] M. Silva and L. Recalde, "On fluidification of Petri net models: from discrete to hybrid and continuous models," *Annual Reviews in Control*, vol. 28, no. 2, pp. 253–266, 2004.

[3] F. Balduzzi, G. Menga, and A. Giua, "First-order hybrid Petri nets: a model for optimization and control," *IEEE Trans. on Robotics and Automation*, vol. 16, no. 4, pp. 382–399, 2000.

[4] L. Recalde, S. Haddad, and M. Silva, "Continuous Petri Nets: Expressive Power and Decidability Issues," in *Proc. of the 5th Int. Symp. on Automated Technology for Verification and Analysis (ATVA2007)*, vol. 4762. Spinger, 2007, pp. 362–377.

[5] L. Recalde and M. Silva, "Petri Nets Fluidification revisited: Semantics and Steady state," *APII-JESA*, vol. 35, no. 4, pp. 435–449, 2001.

[6] M. Silva, E. Teruel, and J. M. Colom, "Linear algebraic and linear programming techniques for the analysis of net systems," in *Lectures in Petri Nets. I: Basic Models*, ser. LNCS, G. Rozenberg and W. Reisig, Eds. Springer, 1998, vol. 1491, pp. 309–373.

[7] L. Habets, P. Collins, and J. van Schuppen, "Reachability and control synthesis for piecewise-affine hybrid systems on simplices," *IEEE Trans. Aut. Control*, vol. 51, pp. 938–948, 2006.

[8] A. Chutinan and B. H. Krogh, "Verification of infinite-state dynamic systems using approximate quotient transition systems," *IEEE Trans. Aut. Control*, vol. 46, no. 9, pp. 1401–1410, 2001.

[9] B. Yordanov, C. Belta, and G. Batt, "Model checking discrete time piesewise affine systems: application to gene networks," in *European Control Conference*, Kos, Greece, 2007.

[10] M. Kloetzer and C. Belta, "Reachability analysis of multi-affine systems," in *Hybrid Systems: Computation and Control: 9th International Workshop*, ser. LNCS, J. Hespanha and A. Tiwari, Eds. Springer Berlin / Heidelberg, 2006, vol. 3927, pp. 348 – 362.

[11] C. Mahulea, L. Recalde, and M. Silva, "Basic Server Semantics and Performance Monotonicity of Continuous Petri Nets," *Discrete Event Dynamic Systems: Theory and Applications*, 2008, to appear.

[12] C. Mahulea, A. Ramírez, L. Recalde, and M. Silva, "Steady state control reference and token conservation laws in continuous Petri net systems," *IEEE Trans. on Autom. Science and Engineering*, vol. 5, no. 2, pp. 307–320, 2008.

[13] E. M. M. Clarke, D. Peled, and O. Grumberg, *Model checking*. MIT Press, 1999.

[14] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Trans. Aut. Control*, vol. 53, no. 1, pp. 287–297, 2008.

[15] L. Lamport, "The temporal logic of actions," *ACM Trans. Program. Lang. Syst.*, vol. 16, no. 3, pp. 872–923, 1994.

[16] G. Holzmann, *The SPIN Model Checker, Primer and Reference Manual*. Reading, Massachusetts: Addison-Wesley, 2004.

[17] C. Belta and L. Habets, "Constructing decidable hybrid systems with velocity bounds," in *43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, 2004.

[18] L. Habets and J. van Schuppen, "A control problem for affine dynamical systems on a full-dimensional polytope," *Automatica*, vol. 40, pp. 21–35, 2004.

[19] T. Murata, "State equation, controllability, and maximal matchings of Petri nets," *IEEE Trans. on Automatic Control*, vol. 22, no. 3, pp. 412–416, 1977.

[20] K. Fukuda, "CDD/CDD+ package," URL http://www.cs.mcgill.ca/∼fukuda/soft/cdd_home/cdd.html, 1997.

[21] M. Kloetzer, "Symbolic motion planning and control," Ph.D. dissertation, Boston University, Boston, MA, 2008.