Nonlinear Analysis: Hybrid Systems 6 (2012) 806–817

Contents lists available at SciVerse ScienceDirect

Nonlinear Analysis: Hybrid Systems

journal homepage: www.elsevier.com/locate/nahs

SimHPN: A MATLAB toolbox for simulation, analysis and design with hybrid Petri nets ☆

Jorge Júlvez *, Cristian Mahulea, Carlos-Renato Vázquez

Aragón Institute for Engineering Research (I3A), University of Zaragoza, Maria de Luna 1, 50018 Zaragoza, Spain

ARTICLE INFO

Article history:
Received 13 July 2011
Accepted 3 October 2011

Keywords:
Hybrid Petri nets
Software tools
Hybrid systems

ABSTRACT

This paper presents a MATLAB embedded package for hybrid Petri nets called SimHPN. It offers a collection of tools devoted to simulation, analysis and synthesis of dynamical systems modeled by hybrid Petri nets. The package supports several server semantics for the firing of both, discrete and continuous, types of transitions. Besides providing different simulation options, SimHPN offers the possibility of computing steady state throughput bounds for continuous nets. For such a class of nets, optimal control and observability algorithms are also implemented. The package is fully integrated in MATLAB which allows the creation of powerful algebraic, statistical and graphical instruments that exploit the routines available in MATLAB.

1. Introduction

Petri nets (PNs) [1,2] is a mathematical formalism for the description of discrete-event systems, that has been successfully used for modeling, analysis and synthesis purposes of such systems. A key feature of a PN is that its structure can capture graphically fundamental primitives in concurrency theory such as parallelism, synchronization, mutual exclusion, etc. The state of a PN system is given by a vector of non-negative integers representing the marking of its places.

As any other formalism for discrete event systems, PNs suffer from the *state explosion problem* which produces an exponential growth of the size of the state space with respect to the initial marking. One way to avoid the state explosion is to relax the integrality constraint in the firing of transitions and deal with transitions that are fired in real amounts. A transition whose firing amount is allowed to be any real number between zero and its enabling degree is said to be a *continuous transition*. The firing of a continuous transition can produce a real, not integer, number of tokens in its input and output places. If all transitions of a net are continuous, then the net is said to be continuous. If a non-empty proper subset of transitions is continuous, then the net is said to be hybrid [3].

Different time interpretations can be considered for the firing of continuous transitions. The most popular ones are infinite and finite server semantics which represent a first order approximation of the firing frequency of discrete transitions. For a broad class of Petri nets, infinite server semantics offer a better approximation of the steady-state throughput than finite server semantics [4]. Moreover, finite server semantics can be exactly mimicked by infinite server semantics in discrete transitions simply by adding a self-loop place. A third firing semantics, called product semantics, is also frequently used when dealing with biochemical and population dynamics systems.

In this paper, we present a new MATLAB embedded software called *SimHPN* that provides support for infinite server and product semantics in both, discrete and continuous, types of transition. A description of a preliminary version of this software can be found in [5,6]. As far as we know, this is the first MATLAB package that enables the analysis and simulation of hybrid

☆ This work has been partially supported by the European Community's Seventh Framework Programme under project DISC (Grant Agreement n. INFSO-ICT-224498), by CICYT–FEDER grants DPI2010-20413 and TIN2007-66523 and by Fundación Aragón I + D.

* Corresponding author. Tel.: +34 976762336; fax: +34 976761914.
E-mail addresses: julvez@unizar.es (J. Júlvez), cmahulea@unizar.es (C. Mahulea), cvazquez@unizar.es (C.-R. Vázquez).

1751-570X/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.nahs.2011.10.001

nets with these two firing semantics. There already exists a toolbox dealing with discrete Petri nets [7], and one for the so-called first order hybrid Petri nets [8] which provides support for continuous transitions under finite server semantics. The main features of the *SimHPN* toolbox are (i) simulation of hybrid Petri nets under different server semantics; (ii) computation of steady state throughput bounds; (iii) computation of minimal P–T *semiflows*; (iv) optimal sensor placement; (v) optimal control algorithm; (vi) import models from different graphical Petri net editors.

The paper is organized as follows. Section 2 introduces the formal definition of the hybrid Petri nets supported by *SimHPN*. Section 3 briefly presents the main features of the package. Sections 4–6 exemplify those features by applying them to three case studies. Section 7 concludes the paper.

2. Hybrid Petri nets

Hybrid Petri nets [3,9] represent a powerful modeling formalism that allows the integration of both continuous and discrete dynamics in a single net model. This section defines the class of hybrid nets supported by *SimHPN*. In the following, the reader is assumed to be familiar with Petri nets (PNs) (see [1,2] for a gentle introduction).

2.1. Untimed hybrid Petri net systems

Definition 1. A *Hybrid Petri Net (HPN) 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*, with set of places P, set of transitions T, pre and post incidence matrices $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{R}_{\geq 0}^{|P|\times|T|}$, and $\boldsymbol{m}_0 \in \mathbb{R}_{\geq 0}^{|P|}$ is the *initial marking*.

The token load of the place p_i at marking $\boldsymbol{m}$ is denoted by m_i and the *preset* and *postset* of a node $X \in P \cup T$ are denoted by $^\bullet X$ and $X^\bullet$, respectively. For a given incidence matrix, e.g., $\boldsymbol{Pre}$, $\boldsymbol{Pre}(p_i, t_j)$ denotes the element of $\boldsymbol{Pre}$ in row i and column j.

In an HPN, the set of transitions T is partitioned in two sets $T = T^c \cup T^d$, where T^c contains the set of continuous transitions and T^d the set of discrete transitions. In contrast to other works, the set of places P is not explicitly partitioned, i.e., the marking of a place is a natural or real number depending on the firings of its input and output transitions. Nevertheless, in order to make net models easier to understand, those places whose marking can be a real non-integer number will be depicted as double circles (see p_1^1 in Fig. 3), and the rest of places will be depicted as simple circles (such places will have integer markings; see p_5^1 in Fig. 3). Continuous transitions are graphically depicted as two bars (see t_4^1 in Fig. 3), while discrete transitions are represented as empty bars (see t_5^1 in Fig. 3).

Two enabled transitions t_i and t_j are in conflict when they cannot occur at the same time. For this, it is necessary that $^\bullet t_i \cap {}^\bullet t_j \neq \emptyset$, and in that case it is said that t_i and t_j are in structural conflict relation. Right and left non negative annullers of the token flow matrix $\boldsymbol{C}$ are called T- and P-*semiflows*, respectively. A semiflow $\mathbf{v}$ is *minimal* when its support, $\|\mathbf{v}\| = \{i \mid \mathbf{v}(i) \neq 0\}$, is not a proper superset of the support of any other semiflow, and the greatest common divisor of its elements is one. If there exists $\boldsymbol{y} > 0$ such that $\boldsymbol{y} \cdot \boldsymbol{C} = 0$, the net is said to be *conservative*, and if there exists $\boldsymbol{x} > 0$ satisfying $\boldsymbol{C} \cdot \boldsymbol{x} = 0$, the net is said to be *consistent*. As it will be seen, the basic tasks that *SimHPN* can perform on untimed hybrid Petri nets are related to the computation of minimal T- and P-semiflows.

The enabling degree of a transition $t_j \in T$ is:

$$\text{enab}(t_j, \boldsymbol{m}) = \begin{cases} \min\limits_{p_i \in {}^\bullet t_j} \left\lfloor \dfrac{m_i}{\boldsymbol{Pre}(p_i, t_j)} \right\rfloor & \text{if } t_j \in T^d \\ \min\limits_{p_i \in {}^\bullet t_j} \dfrac{m_i}{\boldsymbol{Pre}(p_i, t_j)} & \text{if } t_j \in T^c. \end{cases} \tag{1}$$

Transition $t_j \in T$ is *enabled* at $\boldsymbol{m}$ iff $\text{enab}(t_j, \boldsymbol{m}) > 0$. An enabled transition $t_j \in T$ can fire in any amount α such that $0 \leq \alpha \leq \text{enab}(t_j, \boldsymbol{m})$ and $\alpha \in \mathbb{N}$ if $t_j \in T^d$ and $\alpha \in \mathbb{R}$ if $t_j \in T^c$. Such a firing leads to a new marking $\boldsymbol{m}' = \boldsymbol{m} + \alpha \cdot \boldsymbol{C}(\cdot, t_j)$, where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the token-flow matrix and $\boldsymbol{C}(\cdot, t_j)$ is its j column. If $\boldsymbol{m}$ is reachable from $\boldsymbol{m}_0$ through a finite sequence σ, the *state (or fundamental) equation*, $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$ is satisfied, where $\boldsymbol{\sigma} \in \mathbb{R}_{\geq 0}^{|T|}$ is the firing count vector. According to this firing rule the class of nets defined in Definition 1 is equivalent to the class of nets defined in [3,9].

2.2. Timed hybrid Petri net systems

Different time interpretations can be associated to the firing of transitions. Once an interpretation is chosen, the state equation can be used to show the dependency of the marking on time, i.e., $\boldsymbol{m}(\tau) = \boldsymbol{m_0} + \boldsymbol{C} \cdot \boldsymbol{\sigma}(\tau)$. The term $\boldsymbol{\sigma}(\tau)$ is the firing count vector at time τ. Depending on the chosen time interpretation, the firing count vector $\sigma_j(\tau)$ of a transition $t_j \in T^c$ is differentiable with respect to time, and its derivative $f_j(\tau) = \dot{\sigma}_j(\tau)$ represents the *continuous flow* of t_j. As for the timing of discrete transitions, several definitions exist for the flow of continuous transitions. *SimHPN* accounts for infinite server and product server semantics in both continuous and discrete transitions, and additionally, discrete transitions are also allow to have deterministic delays.

Definition 2. A *Timed Hybrid Petri Net (THPN) system* is a tuple $\langle \mathcal{N}, \boldsymbol{m}_0, \mathit{Type}, \boldsymbol{\lambda}\rangle$ where $\langle \mathcal{N}, \boldsymbol{m}_0\rangle$ is a HPN, $\mathit{Type} : T \to \{id, pd, dd, ic, pc\}$ establishes the time semantics of transitions and $\boldsymbol{\lambda} : T \to \mathbb{R}_{\geq 0}$ associates a real parameter to each transition related to its semantics.