First-order hybrid Petri nets. An application to distributed manufacturing systems

M. Dotoli [a] M.P. Fanti [a] A. Giua [b] C. Seatzu [b]

[a] *Dip. di Elettrotecnica ed Elettronica, Politecnico di Bari, Italy*
Email: {dotoli,fanti}@deemail.poliba.it

[b] *Dip. di Ing. Elettrica ed Elettronica, Università di Cagliari, Italy*
Email: {giua,seatzu}@diee.unica.it

Abstract

[1] In this paper we consider Hybrid Petri Nets (HPNs), a particular formalism that combines fluid and discrete event dynamics. We first provide a survey of the main HPNs models that have been presented in the literature in the last decades. Then, we focus on a particular HPN model, namely the First-Order Hybrid Petri Net (FOHPN) model, whose continuous dynamics are piece-wise constant. Here the problem of designing an optimal controller simply requires solving on-line an appropriate linear integer programming problem. In this paper we show how FOHPNs can efficiently represent the concurrent activities of Distributed Manufacturing Systems (DMS), and some interesting optimization problems are also solved via numerical simulation.

Key words: Hybrid Petri nets; first-order hybrid Petri nets; manufacturing systems; distributed manufacturing systems.

1 Introduction

Petri Nets (PNs) are a discrete event model firstly proposed by Carl Adam Petri in his PhD thesis in the early sixties (Petri, 1962). The main feature of a (discrete) PN is that its state is a vector of non-negative integers. This is a major advantage with respect to other formalisms such as automata, where the state space is a symbolic unstructured set, and has been exploited to develop many analysis techniques that do not require to enumerate the state space (structural analysis) (Colom and Silva, 1990). Another key feature of PNs is their capacity to graphically represent and visualize primitives such as parallelism, concurrency, synchronization, mutual exclusion, etc.

[1] Published as: M. Dotoli, M.P. Fanti, A. Giua, C. Seatzu, "First-Order Hybrid Petri nets. An application to distributed manufacturing systems," *Nonlinear Analysis: Hybrid Systems*, Vol. 2., No. 2, pp. 408-430, June 2008.

In the related literature various PNs extensions have been proposed. In this paper we focus on *Continuous* and *Hybrid* PNs.

Continuous Petri Nets (CPNs) originate from the requirement of reducing the computational complexity of the analysis and optimization of realistic scale problems, that quickly become analytically and computationally intractable with discrete event models. As an example, this often occurs in many production systems where the number of parts in a buffer may be very large, thus it may be convenient to approximate it with a real number. In effect, the idea of "fluidification" has been firstly applied to other discrete event models, such as automata. Later, in 1987 it has also been successfully applied to PNs by David and Alla (David and Alla, 1987). The main advantages in using fluid approximations for analysis and control of complex systems can be summarized as follows. First, there is the possibility of considerable increase in computational efficiency, because the simulation of fluid models can often be performed much more efficiently than their discrete counterpart. Second, fluid approximations provide an aggregate formulation to deal with complex systems, thus reducing the dimension of the state space. The resulting simple structures allow explicit computation and performance optimization. Third, the design parameters in fluid models are continuous, hence it is possible to use gradient information to speed up optimization and to perform sensitivity analysis. Furthermore, in many cases it has also been shown that fluid approximations do not introduce significant errors when carrying out performance analysis via simulation.

In general different fluid approximations are necessary to describe the same system, depending on its discrete state, e.g., in the manufacturing domain, machines working or down, buffers full or empty, and so on. Thus, the resulting model can be better described as a *hybrid model*, where a different continuous dynamics is associated to each discrete state.

Hybrid Petri Nets (HPNs) keep all those good features that make discrete PNs a valuable discrete-event model: they do not require the exhaustive enumeration of the state space and can finitely describe systems with an infinite state space; they allow modular representation where the structure of each module is kept in the composed model; the discrete state is represented by a vector and not by a symbolic label, thus linear algebraic techniques may be used for their analysis.

Different HPN models have been proposed in the literature, but there is so far no widely accepted classification of such models.

1.1 Structure of the paper

A survey of the the most important HPN models is given in Section 2.

Then, in Section 3 we focus our attention on a particular model of HPNs, called *First–Order Hybrid Petri Nets* (FOHPNs) because its continuous dynamics are piece-wise constant. FOHPNs were originally proposed in (Balduzzi *et al.*, 2000), and have been efficiently used in many application domains, such as manufacturing systems (Balduzzi *et al.*, 2001*b*; Giua *et al.*, 2001), and inventory control (Furcas *et al.*, 2001). Interesting optimization problems have also been studied considering real applications, such as a bottling plant (Giua *et al.*, 2001) and a cheese factory (Furcas *et al.*, 2001).

Several problems related to the control and analysis of FOHPNs have been studied (Balduzzi *et al.*, 2000; Balduzzi *et al.*, 2001*a*); they are briefly summarized in Section 4, where we also underline our future lines of research in this topic.

In the last part of this paper (see Section 5) we also show how FOHPN can be efficiently used for modeling and controlling large and complex systems such as *Distributed Manufacturing Systems* (DMS). DMS are complex emerging paradigms whose analysis, design and management is currently an active area of research (Dotoli *et al.*, 2006*a*; Viswanadham and Raghavan, 2000; Viswanadham and Gaonkar, 2003). More precisely, a DMS is defined as a collection of independent companies possessing complementary skills and integrated with transportation and storage systems, information and financial flows, with all entities collaborating to meet the market demand (Viswanadham and Gaonkar, 2003). Appropriate modeling and analysis of such highly complex systems are crucial for performance evaluation and to compare competing DMS. However, in the related literature very few contributions deal with the problem of modelling and analyzing the DMS operational behaviour. Viswanadham and Raghavan (2000) model DMS as Discrete Event Dynamical Systems in which the evolution depends on the interaction of discrete events such as the arrival of the components at the facilities, the departures of the transports, the starts of the operations at the manufacturers and the assemblers. In (Desrochers *et al.*, 2005) a two product DMS is modelled by *complex-valued token* PNs and the performance measures are determined by simulation. However, the limit of such formalisms is the modelling of products or batches of parts by means of discrete quantities (i.e., tokens). This assumption is not realistic in large DMS with a huge amount of material flow. Hence, this paper uses FOHPN to model and manage DMS (Dotoli *et al.*, 2006*b*). Using a modular approach based on the idea of the bottom-up methodology (Zhou and Venkatesh, 1998), the paper develops a modular FOHPN model of DMS. In particular, transporters and manufacturers are described by continuous transitions, buffers are continuous places, and products are represented by continuous flows (fluids) routing from manufacturers, buffers and transporters.

In Section 5 an example of DMS is considered and, applying a *Make-To-Stock* management policy and appropriate inventory control rules (Viswanadham and Raghavan, 2000), the system is analyzed under different operative conditions.

1.2 Main contributions

The original contribution of this paper is threefold.

Firstly, we provide a detailed survey of the different HPN models that have been proposed in the literature. The main theoretical results and the main application areas within each framework are also mentioned.

Secondly, we show how FOHPN can be efficiently used to model DMS.

Finally, we show how the resulting model leads to easy simulation and to easy performance analysis and optimization via different scenarios.

2 Hybrid Petri nets

A unified classification of all HPN models so far appeared in the literature is still missing. The goal of this section is that of providing, following (Giua, 2006), a survey of the main references in the field of HPNs. Other interesting surveys are given in (Antsaklis and Koutsoukos, 1998; David, 1997; David and Alla, 2005).

The first fluid PN model is the so called "Continuous and Hybrid Petri net" model introduced by David and Alla in their seminal paper (David and Alla, 1987). Based on this first formalism, a family of extended hybrid models have been proposed, and the main contributions in this area are reported in Section 2.1.

Different formalism, motivated by particular applications, have also been presented in the literature. In Section 2.2 we review some of them: *Fluid Stochastic Petri nets*, *Batch nets*, *DAE-Petri nets*, *Hybrid Flow nets*, *Differential Petri nets* and *High-Level Hybrid nets*.

There are also other interesting approaches dealing with HPNs that are not discussed here for sake of brevity; a list of them can be found in the website (Giua, 2006).

2.1 Continuous and Hybrid Petri nets

All the works collected under this heading are based or directly inspired on the model presented by R. David and H. Alla in the late eighties (David and Alla, 1987). These authors have obtained a *continuous model* by *fluidification*, i.e., by relaxing the condition that the marking be an integer vector. *Hybrid Petri nets* are then made of a "continuous part" (continuous places and transitions) and a "discrete part" (discrete places and transitions). The continuous part can model systems with continuous flows and the discrete part models the logic functioning. In particular, the constant-speed HPN model of David and Alla (Alla and David, 1998*b*) derives from deterministic timed PNs.

Several contributions in this framework have been presented in the last decade, as well as some interesting extensions with respect to the original model. A discussion on these issues is provided in the rest of the subsection.

2.1.1 Steady-state, controllability and analysis

The problem of determining an optimal stationary mode of operation for a system described by a timed CPN has been studied in (Gaujal and Giua, 2004). Here the authors solve conflicts at places by using stationary routing parameters, and show how to compute the stationary firing rate for all transitions via linear programming, so as to determine the optimal routing parameters that maximize user-defined functions of the firing rates.

Some characterizations of equilibrium points in steady-state are given in (Mahulea *et al.*, 2006*b*), where an optimal steady-state control is also studied. In particular, the authors prove that, under the assumptions that all transitions are controllable, the control problem can be solved in polynomial time.

An interesting comparison on two different techniques to compute the steady-state of continuous nets was made in (Demongodin and Giua, 2002): a method based on linear programming and a method based on graph theory.

Other interesting papers have been devoted to the problem of production frequencies estimation for systems that are modeled by CPNs (Lefebvre, 2000*b*; Lefebvre, 2000*a*), to the design of observers (Júlvez *et al.*, 2004), to the reachability analysis (Júlvez *et al.*, 2003), to the stability analysis (Zerhouni, 2001), and to the deadlock-freeness analysis (Júlvez *et al.*, 2002).

Less recent results related to modeling, analysis and of CPNs have been presented in (Ait-Yahia *et al.*, 1995; Alla, 1995; Alla and David, 1998*a*; Amer-Yahia *et al.*, 1995; Amer-Yahia *et al.*, 1996; Ait-Yahia *et al.*, 1995; Amrah *et al.*, 1996; David and Alla, 1993; Dubois and Alla, 1993; Dubois *et al.*, 1994; Komenda *et al.*, 1998; Bail *et al.*, 1992; Bail *et al.*, 1993).

2.1.2 Optimal control

The problem of deriving an optimal control law for CPNs under the assumption of *finite servers semantics* has been studied in (Bemporad *et al.*, 2004). Here the proposed approach consists in transforming the CPN model into an equivalent hybrid system whose evolution is described by means of discrete-event steps (each step coincides with the occurrence of an event in the CPN). It is shown how to design a mixed integer linear programming problem in order to compute the optimal control solution of different performance criteria.

In (Mahulea *et al.*, 2006*a*) the authors considered timed CPNs under *infinite servers semantics* that usually provide a much better approximation of the discrete system than finite servers semantics (Mahulea *et al.*, 2006*c*). They dealt with the problem of controlling CPNs in order to reach a final (steady state) configuration while minimizing a quadratic performance index. In particular, they considered CPNs subject to external control actions, where the only admissible control law consists in slowing down the firing speed of transitions (Silva and Recalde, 2004). The formulation of a discrete-time *linear positive* model with dynamic (or state-based) constraints on the control input, enables one to design both a control law based on *implicit model predictive control*, and a *state-feedback* control law based on *explicit model predictive* *control* (Bemporad *et al.*, 2002).

2.1.3 Applications

CPNs have been mainly applied in the manufacturing domain, see e.g. (Alla and David, 1988; Allam and Alla, 1997; Allam and Alla, 1998; Amrah *et al.*, 1997*b*; Amrah *et al.*, 1997*a*; Amrah *et al.*, 1998*a*; Amrah *et al.*, 1998*b*; El-Fouly *et al.*, 1998; Lefebvre, 1999; Zerhouni and Alla, 1990; Zerhouni and Alla, 1992), even if some other interesting applications have been presented, like (Amer-Yahia *et al.*, 1997) dealing with biological systems, and (Júlvez and Boel, 2005) dealing with transportation systems.

2.1.4 First-Order Hybrid Petri nets

First-Order Hybrid Petri Nets (FOHPNs) follow the formalism described by R. David and H. Alla (Alla and David, 1998*b*) with the addition of algebraic analysis techniques, and have been firstly presented by F. Balduzzi, A. Giua and G. Menga in (Balduzzi *et al.*, 2000).

FOHPNs consist of continuous places holding fluid, discrete places containing a non-negative integer number of tokens, and transitions, either discrete or continuous. As in all hybrid models, in FOHPNs the authors distinguish two behavioral levels: time-driven and event-driven. The continuous time-driven evolution of the net is described by first-order fluid models, i.e., models in which the continuous flows have constant rates and the fluid content of each continuous place varies linearly with time. A discrete-event model describes the behaviour of the net that, upon the occurrence of macro-events, evolves through a sequence of macro-states. The authors set up a linear algebraic formalism to study the first–order continuous behavior of this model and show how its control can be framed as a conflict resolution policy that aims to optimize a given objective function. The use of linear algebra leads to sensitivity analysis that allows one to study of how changes in the structure of the model influence the optimal behavior.

This model is extensively presented in the rest of this paper.

2.2 Other models

2.2.1 Fluid Stochastic Petri nets

The *Fluid Stochastic Petri Net* (FSPN) model has been firstly presented by K.S. Trivedi and V.G. Kulkarni in the early nineties (Trivedi and Kulkarni, 1993) and further elaborated in (Horton *et al.*, 1998). In (Trivedi and Kulkarni, 1993) the authors extend the stochastic Petri nets framework (Marsan *et al.*, 1995*a*) to FSPNs by introducing places with continuous tokens and arcs with fluid flow so as to handle stochastic fluid flow systems. No continuous transitions are present in this model, and the set of transitions is partitioned in timed transitions and immediate transitions, where timed transitions have an exponentially distributed firing time. They define hybrid nets in such a way that the discrete and continuous portions may affect each other.

A new feature called *flush-out arc* has been added in (Gribaudo *et al.*, 2001): a flush-out arc connects a timed transition to a fluid place and has the effect of "instantaneously" emptying the fluid place as the transition fires, thus enabling discrete jumps in the fluid level. In (Gribaudo *et al.*, 2001) it is clearly shown that flush-out arcs considerably increase the modeling power of FSPNs. Despite this and the major complexity of the new model, an analytical description of the stochastic marking process is feasible, and the authors in (Gribaudo *et al.*, 2001) also indicated a general procedure to automatically derive the solution equations from the model specification.

Other interesting papers in this framework are (Ciarlo *et al.*, 1999; Horton *et al.*, 1998; Nicol and Miner, 1995). In particular, in (Ciarlo *et al.*, 1999) the authors describe a general method for the simulation of FSPNs that results in a rather complex set of partial differential equations. Wolter and Hommel also extend the fluid stochastic Petri net model to second order flow approximations (Wolter, 1997; Wolter and Hommel, 1997; Wolter and Hommel, 1998).

2.2.2 Batch Petri nets

In a *batch process*, the material is operated by finite quantities (the batches). At any time, an integer number of batches are in operation at certain locations in the plant, thus the production process behaves like a discrete manufacturing system where a batch of fluid in a buffer or a machine can be considered as a part. The characterization of a batch implies real numbers describing the amount of material, and some operating conditions like temperature or pressure. The material is continuously transferred from different equipments, and during the transfer the discrete nature of the batch temporarily disappears. Thus the mathematical model of a batch process has to be hybrid (Champagnat *et al.*, 2001). An overview of Petri net modeling techniques for batch systems is given in (Champagnat *et al.*, 2001).

There is however, a formalism called *Batch Petri nets* (BPN), that has been derived by I. Demongodin, N. Audry and F. Prunet (Demongodin *et al.*, 1993; Demongodin *et al.*, 1998) as a modeling tool for this particular class of processes. It intends to model variable delays on continuous flows by adding to a hybrid Petri net special nodes called *batch nodes*. Batch nodes combine both a discrete event and a linear continuous dynamic behaviour in a single structure. Evolution rules are determined in order to do the simulation of systems based on accumulation phenomena, thus resulting well suited to model high throughput production lines (Audry *et al.*, 1994; Audry and Prunet, 1995; Demongodin, 1999). Extensions of the basic model have been proposed later in (Demongodin, 2001), where *Generalized* BPN have been introduced that enable to represent phenomena as synchronization, parallelism and proportion of batches or set of batches, thus enlarging the modeling power of BPN.

2.2.3 DAE-Petri nets

Differential Algebraic Equations-Petri nets (DAE-PNs) are based on the model presented by D. Andreu, J.C. Pascal, C. Valentin-Roubinet, and R. Valette (Andreu *et al.*, n.d.; Andreu *et al.*, 1995; Andreu *et al.*, 1996; Champagnat *et al.*, 1998; Daubas *et al.*, 1994; Valentin-Roubinet, 1994*a*; Valentin-Roubinet, 1994*b*; Valentin-Roubinet, 1998; Vibert *et al.*, 1997). This approach does not try to represent in a unified way the continuous and discrete aspects, as it is the case in HPNs in which there are discrete places with an integer token load and continuous ones with a real token load. On the contrary, the model focuses on the interaction between a discrete Petri net model that captures the discrete behaviour of a batch system, and a continuous model which is a set of differential algebraic equations. DAE-PNs can be seen as an extension of hybrid automata (Alur *et al.*, 1993; Puri and Varaiya, 1996). This approach is well suited for modelling batch processes where it is necessary to concurrently deal with continuous and discrete models. It has also been tested in the food industry for the validation of scheduling policies and has been developed for supervisory control and reactive scheduling.

2.2.4 Hybrid Flow nets

Hybrid Flow nets (HFNs) have been proposed by J.-M. Flaus (Flaus, 1996; Flaus, 1997; Flaus and Ollagnon, 1997; Flaus and Alla, 1997). This approach is based on the analysis of a system as a set of continuous and discrete flows. The notion of HFNs can then be seen as an extension of Petri nets for hybrid systems. This modeling tool is made of a continuous flow net interacting with a Petri net according to a control interaction, that is to say the Petri net controls the continuous flow net and vice versa. The overall philosophy of Petri nets is preserved again. The discrete part is a Petri net while the continuous part is called *continuous flow* net whose dynamic evolution has be defined so that to be similar to the one of Petri nets, with a continuous enabling rule and a continuous firing rule. HFNs are well suited for the modeling and control of industrial transformation processes for which the dynamics behavior has an hybrid nature.

2.2.5 Differential Petri nets

Differential Petri Nets (DPNs) have been firstly presented by I. Demongodin and N.T. Koussoulas in (Demongodin and Koussoulas, 1998). The main feature of this class of Petri nets is that it allows us to model continuous-time dynamic processes represented by a finite number of linear first-order differential state equations. The differential Petri net is defined through the introduction of a new kind of place and transition, namely, the *differential place* and the *differential transition*. The marking of the differential place represents a state variable of the continuous system that is modeled. To every differential transition it is associated a firing speed representing either a variable proportional to a state variable or an independent variable. A differential transition is always enabled, thus to discretize the continuous system, to any differential transition is associated a firing frequency representing the integration step that would be used when carrying out an integration of the differential equation. Evolution rules have been developed to precise the simulation of hybrid systems composed by a continuous part cooperating with a discrete event part, i.e., the typical paradigm of a supervisory control system. DPNs can integrate all kinds of discrete Petri nets. In fact, the introduction of the new type of places and of transitions does not change the description and the evolution of the discrete part. Thus, it is possible to integrate in DPNs, stochastic, interpreted and many other kinds and extensions of Petri nets. Through the introduction of this new formalism it is possible to model concurrently discrete-event processes and continuous-time dynamic processes, represented by systems of linear ordinary differential equations. This model can contribute to the performance analysis and design of industrial supervisory control systems and of hybrid control systems in general.

2.2.6 High-Level Hybrid Petri Nets

The works collected under this heading consider different models presented by several authors (Chen and Hanisch, 1998; Genrich and Schuart, 1998; Giua and Usai, 1998). All these models, however, are based on High-Level nets, i.e., nets characterized by the use of *structured individual tokens*. *High-Level Hybrid Petri Nets* (HLHPNs) are a useful model that provides a simple graphical representation of hybrid systems and takes advantage of the modular structure of Petri nets in giving a compact description of systems composed of interacting subsystems, both time-continuous and discrete-event. The use of colors in the continuous places allows one to model continuous variables that may take negative values.

3 First-order hybrid Petri nets

In this section we provide a detailed presentation of the FOHPN model (Balduzzi *et al.*, 2000). For a more comprehensive introduction to place/transition Petri nets see (Murata, 1989).

3.1 Net structure

A FOHPN is a structure

$$N = (P, T, Pre, Post, \mathcal{D}, \mathcal{C}).$$

The set of *places* $P = P_d \cup P_c$ is partitioned into a set of *discrete* places P_d (represented as circles) and a set of *continuous* places P_c (represented as double circles). The cardinality of P, P_d and P_c is denoted n, n_d and n_c. We assume that the place labeling is such that: $P_c = \{p_i \mid i = 1, \ldots, n_c\}$, $P_d = \{p_i \mid i = n_c + 1, \ldots, n\}$.

The set of *transitions* $T = T_d \cup T_c$ is partitioned into a set of discrete transitions T_d and a set of continuous transitions T_c (represented as double boxes). The set $T_d = T_I \cup T_D \cup T_E$ is further partitioned into a set of *immediate* transitions T_I (represented as bars), a set of *deterministic timed* transitions T_D (represented as black boxes), and a set of *exponentially distributed timed* transitions T_E (represented as white boxes). The cardinality of T, T_d and T_c is denoted q, q_d and q_c. We also denote with q_t the cardinality of the set of timed transitions $T_t = T_D \cup T_E$. We assume that the transition labeling is such that: $T_c = \{t_j \mid j = 1, \ldots, q_c\}$, $T_t = \{t_j \mid j = q_c + 1, \ldots, q_c + q_t\}$, $T_I = \{t_j \mid j = q_c + q_t + 1, \ldots, q\}$.

The *pre-* and *post-incidence functions* that specify the arcs are (here $\mathbb{R}_0^+ = \mathbb{R}^+ \cup \{0\}$):

$$Pre, Post : \begin{cases} P_c \times T \to \mathbb{R}_0^+ \\ P_d \times T \to \mathbb{N} \end{cases} .$$

We require (*well-formed nets*) that for all $t \in T_c$ and for all $p \in P_d$, $Pre(p,t) = Post(p,t)$. This ensures that the firing of continuous transitions does not change the marking of discrete places.

The function $\mathcal{D} : T_t \to \mathbb{R}^+$ specifies the timing associated to timed discrete transitions. We associate to a deterministic timed transition $t_j \in T_D$ its (constant) firing delay $\delta_j = \mathcal{D}(t_j)$. We associate to an exponentially distributed timed transition $t_j \in T_E$ its average firing rate $\lambda_j = \mathcal{D}(t_j)$, i.e. the average firing delay is $1/\lambda_j$, where λ_j is the parameter of the corresponding exponential distribution.

The function $\mathcal{C} : T_c \to \mathbb{R}_0^+ \times \mathbb{R}_\infty^+$ specifies the *firing speeds* associated to continuous transitions (here $\mathbb{R}_\infty^+ = \mathbb{R}^+ \cup \{\infty\}$). For any continuous transition $t_j \in T_c$ we let $\mathcal{C}(t_j) = (V'_j, V_j)$, with $V'_j \leq V_j$. Here V'_j represents the *minimum firing speed* (mfs) and V_j represents the *maximum firing speed* (MFS). In the following, unless explicitly specified, the mfs of a continuous transition t_j will be $V'_j = 0$.

We denote the preset (postset) of transition t as ${}^\bullet t$ ($t^\bullet$) and its restriction to continuous or discrete places as ${}^{(d)}t = {}^\bullet t \cap P_d$ or ${}^{(c)}t = {}^\bullet t \cap P_c$. Similar notation may be used for presets and postsets of places. The *incidence matrix* of the net is defined as $\boldsymbol{C}(p,t) = Post(p,t) - Pre(p,t)$. The restriction of $\boldsymbol{C}$ to P_X and T_Y ($X, Y \in \{c, d\}$) is denoted $\boldsymbol{C}_{XY}$. Note that by the well-formedness hypothesis $\boldsymbol{C}_{dc} = \boldsymbol{0}_{n_d \times q_c}$.

Example 3.1 Consider the net in Fig. 1.a. Place p_1 is a continuous place. Places p_2, p_3, p_4, p_5 are discrete places. Transitions t_1 and t_2 are continuous transitions with MFS V_1 and V_2; we have not specified the mfs of the continuous transitions because in this case their value is zero. We assume $V_1\, a < V_2\, b$ (here a and b are the arc weights given by Pre and $Post$). Discrete transitions t_3, t_4, t_5, t_6 are exponentially distributed timed transitions whose average firing rates are λ_3, λ_4, λ_5 and λ_6 respectively.

The two continuous transitions represent two unreliable machines; parts produced by the first machine (t_1) are put in a buffer (place p_1) before being processed by the second machine (t_2). The weight of the arc a (resp., b) represents the ratio between the flow worked by the machine and the flow put into

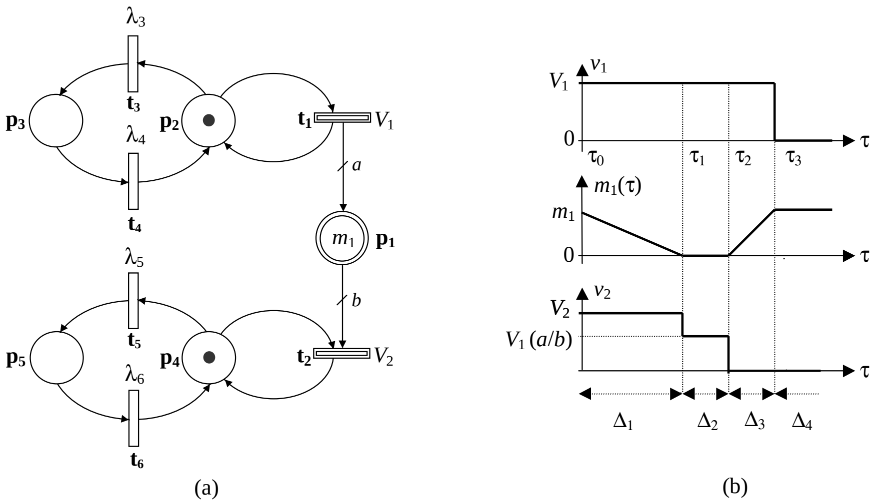

Fig. 1. A First–Order Hybrid Petri Net (a) and its evolution (b).

(resp., taken from) the buffer.

The incidence matrix of this net is

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_{cc} & \boldsymbol{C}_{cd} \\ \boldsymbol{C}_{dc} & \boldsymbol{C}_{dd} \end{bmatrix} = \left[\begin{array}{cc|cccc} a & -b & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{array}\right].$$

We have (well-formedness) $\boldsymbol{C}_{dc} = \boldsymbol{0}_{4\times 2}$. In this particular example we also have $\boldsymbol{C}_{cd} = [0,\ 0,\ 0,\ 0]$. ■

3.2 Marking and enabling

A *marking*

$$\boldsymbol{m} : \begin{cases} P_c \to \mathbb{R}_0^+ \\ P_d \to \mathbb{N} \end{cases}$$

is a function that assigns to each discrete place a non-negative integer number of tokens, represented by black dots, and assigns to each continuous place a fluid volume; m_i denotes the marking of place p_i. The value of the marking at time τ is denoted $\boldsymbol{m}(\tau)$. The restriction of $\boldsymbol{m}$ to P_d and P_c are denoted with $\boldsymbol{m}^d$ and $\boldsymbol{m}^c$, respectively. An *FOHPN system* $\langle N, \boldsymbol{m}(\tau_0)\rangle$ is an FOHPN N with an initial marking $\boldsymbol{m}(\tau_0)$.

The enabling of a discrete transition depends on the marking of all its input places, both discrete and continuous.

Definition 3.2 *Let* $\langle N, \boldsymbol{m} \rangle$ *be an FOHPN system. A discrete transition* t *is* enabled *at* $\boldsymbol{m}$ *if for all* $p_i \in {}^\bullet t$, $m_i \geq Pre(p_i, t)$. ■

A continuous transition is enabled only by the marking of its input discrete places. The marking of its input continuous places, however, is used to distinguish between strongly and weakly enabling.

Definition 3.3 *Let* $\langle N, \boldsymbol{m} \rangle$ *be an FOHPN system. A continuous transition* t *is* enabled *at* $\boldsymbol{m}$ *if for all* $p_i \in {}^{(d)}t$, $m_i \geq Pre(p_i, t)$.

We say that an enabled transition $t \in T_c$ *is:*

- strongly enabled *at* $\boldsymbol{m}$ *if for all places* $p_i \in {}^{(c)}t$, $m_i > 0$;
- weakly enabled *at* $\boldsymbol{m}$ *if for some* $p_i \in {}^{(c)}t$, $m_i = 0$. ■

Example 3.4 In the net in Fig. 1.a the discrete part of the net represents the failure model of the machines. When place p_2 is marked, transition t_1 is enabled, i.e. the first machine is operational; when place p_3 is marked, transition t_1 is not enabled, i.e. the first machine is down. A similar interpretation applies to the second machine. The marking represented in the net shows that initially both machines are operational and the buffer contains a fluid quantity m_1. Transition t_1 is strongly enabled. Transition t_2 is strongly (resp., weakly) enabled if $m_1 > 0$ (resp., $m_1 = 0$). ■

3.3 Net dynamics

We now describe the dynamics of an FOHPN. First, we consider the behaviour associated to discrete transitions that combines a continuous dynamics associated to the timers, and a discrete–event dynamics associated to the transition firing. Then we consider the time–driven behaviour associated to the firing of continuous transitions.

Note that the evolution of an FOHPN is characterized by the occurrence of some events that we call *macro-events*, while the time interval between two consecutive macro-events is called a *macro-period*. As discussed in detail in the following two paragraphs, macro-events may be either related to the firing and/or the enabling condition of discrete transitions, or to the enabling condition and/or the enabling state of a continuous transition.

In the following we will use $\boldsymbol{e}_{i,r}$ to denote the i–th canonical basis vector of dimension r, i.e. the vector

$$\boldsymbol{e}_{i,r} = [\, \underbrace{\overbrace{0,\, \ldots,\, 0,\, 1}^{i},\, 0,\, \ldots,\, 0}_{r} \,]^T.$$

We also define, to simplify the notation, the index $\varrho(j) = j - q_c$ that will be used to define the firing vector associated to a discrete transition.

Discrete transitions dynamics

We associate to each timed transition $t_j \in T_t$ a timer ν_j.

Definition 3.5 (Timers evolution) *Let $\langle N, \boldsymbol{m} \rangle$ be an FOHPN system and $[\tau_k, \tau)$ be an interval of time in which the enabling state of a transition $t_j \in T_t$ does not change. If t_j is enabled in this interval then*

$$\nu_j(\tau) = \nu_j(\tau_k) + (\tau - \tau_k), \tag{1}$$

while if t_j is not enabled in this interval then

$$\nu_j(\tau) = \nu_j(\tau_k) = 0. \tag{2}$$

Whenever t_j is disabled or it fires, its timer is reset to 0. ■

With the notation of (Marsan *et al.*, 1995*b*), we are using a *single–server* semantics, i.e. only one timer is associated to each timed transition, and an *enabling–memory* policy, i.e. each timer is reset to 0 whenever its transition is disabled.

The vector of timers associated to timed transitions is denoted

$$\boldsymbol{\nu} = [\nu_{q_c+1}, \nu_{q_c+2}, \cdots, \nu_{q_c+q_t}]^T \in (\mathbb{R}_0^+)^{q_t}.$$

Note that the timer evolution is continuous and linear during a *macro–period* and may change at the occurrence of the following *macro–events*:

(a) a discrete transition fires, thus changing the discrete marking and enabling (or disabling) a timed transition;
(b) a continuous place reaches a fluid level that enables (or disables) a discrete transition.

An enabled timed transition $t_j \in T_t$ fires when the value of its timer reaches a given value $\nu_j(\tau) = \hat{\nu}_j$: we call the $\hat{\nu}_j$'s the *timer set points*. In the case of a deterministic transition, $\hat{\nu}_j = \delta_j$ is the associated delay. In the case of a stochastic transition, $\hat{\nu}_j$ is the current sample of the associated random variable: it is drawn each time the transition is newly enabled. An immediate transition fires as soon as it is enabled, i.e. it can be considered as a deterministic transition with $\hat{\nu}_j = 0$.

Definition 3.6 (Discrete transition firing) *The firing of a discrete transition t_j at $\boldsymbol{m}(\tau^-)$ yields the marking $\boldsymbol{m}(\tau)$ and for each place p it holds $m_p(\tau) = m_p(\tau^-) + Post(p, t_j) - Pre(p, t_j) = m_p(\tau^-) + C(p, t_j)$. Thus we can write*

$$\begin{cases} \boldsymbol{m}^c(\tau) &= \boldsymbol{m}^c(\tau^-) + \boldsymbol{C}_{cd}\boldsymbol{\sigma}(\tau) \\ \boldsymbol{m}^d(\tau) &= \boldsymbol{m}^d(\tau^-) + \boldsymbol{C}_{dd}\boldsymbol{\sigma}(\tau) \end{cases} \tag{3}$$

where $\boldsymbol{\sigma}(\tau) = \boldsymbol{e}_{\varrho(j),q_d}$ is the firing count vector *associated to the firing of transition t_j.* ■

In the above definition we note that, given the transition labeling defined in section 3.1, a transition t_j is the $\varrho(j) - th$ discrete transition, hence, say, $\boldsymbol{C}_{cd}\boldsymbol{e}_{\varrho(j),q_d}$ represents the column of matrix $\boldsymbol{C}_{cd}$ corresponding to transition t_j.

Example 3.7 In the net in Fig. 1.a we assume that the timer vector $\boldsymbol{\nu} = [\nu_3,\ \nu_4,\ \nu_5,\ \nu_6]^T$ is initially set to zero. Discrete transitions t_3 and t_5 are enabled. Let $\hat{\nu}_3 = 3$ and $\hat{\nu}_5 = 5$ be the timer set points at time $\tau = 0$. Thus, at time $\tau = 3$ transition t_3 fires thus moving the token from place p_2 to place p_3, and the timer ν_3 is reset to 0, while $\nu_5 = 2$. Now, transition t_4 is enabled and if we assume $\nu_4 = 4$ then at time $\tau = 3$ the timer vector is equal to $\boldsymbol{\nu} = [0, 4, 2, 0]^T$. ■

Continuous transitions dynamics

The *instantaneous firing speed* (IFS) at time τ of a transition $t_j \in T_c$ is denoted $v_j(\tau)$. We can write the equation which governs the evolution in time of the marking of a place $p_i \in P_c$ as

$$\dot{m}_i(\tau) = \sum_{t_j \in T_c} \boldsymbol{C}(p_i, t_j) v_j(\tau) = \boldsymbol{e}_{i,n_c}^T \boldsymbol{C}_{cc} \boldsymbol{v}(\tau) \tag{4}$$

where $\boldsymbol{v}(\tau) = [v_1(\tau), \ldots, v_{n_c}(\tau)]^T$ is the IFS vector at time τ. Indeed Equation (4) holds assuming that at time τ no discrete transition is fired and that all speeds $v_j(\tau)$ are continuous in τ.

The enabling state of a continuous transition t_j defines its admissible IFS v_j.

- If t_j is not enabled then $v_j = 0$.
- If t_j is strongly enabled, then it may fire with any firing speed $v_j \in [V_j', V_j]$.
- If t_j is weakly enabled, then it may fire with any firing speed $v_j \in [V_j', \overline{V}_j]$, where the upper bound $\overline{V}_j$ on the firing speed is $\overline{V}_j \leq V_j$ and depends on the flow entering the set of input continuous places ${}^{(c)}t_j$ that are empty. In fact, since t_j cannot remove more fluid from any empty input continuous place $\overline{p}$ than the quantity entered in $\overline{p}$ by other transitions.

The computation of the IFS of enabled transitions is not a trivial task. We will set up in the next section a linear–algebraic formalism to do this. Here we simply discuss the net evolution assuming that the IFS are given.

We say that a *macro–event* occurs when:

(a) a discrete transition fires, thus changing the discrete marking and enabling (or disabling) a continuous transition;
(b) a continuous place becomes empty, thus changing the enabling state of a continuous transition from strong to weak.

Definition 3.8 (Continuous transition firing) *Let τ_k and τ_{k+1} be the occurrence times of two consecutive macro–events as defined above; we assume that within the interval of time $[\tau_k, \tau_{k+1})$ the IFS vector is constant and denoted $\boldsymbol{v}(\tau_k)$. The continuous behaviour of an FOHPN for $\tau \in [\tau_k, \tau_{k+1})$ is described by*

$$\begin{cases} \boldsymbol{m}^c(\tau) = \boldsymbol{m}^c(\tau_k) + \boldsymbol{C}_{cc} \boldsymbol{v}(\tau_k)(\tau - \tau_k) \\ \boldsymbol{m}^d(\tau) = \boldsymbol{m}^d(\tau_k). \end{cases} \tag{5}$$

■

Example 3.9 In the net in Fig. 1.a we assume that the timer vector $\boldsymbol{\nu} = [\nu_3,\ \nu_4,\ \nu_5,\ \nu_6]^T$ is initially set to zero. If $m_1 > 0$ at time τ_0, transitions t_1 and t_2 are strongly enabled and may fire at their maximum speeds, i.e. we choose $v_1 = V_1$ and $v_2 = V_2$. The continuous marking of the net during this macro–period is given, as in Equation 5, by $\boldsymbol{m}^c(\tau) = m_1(\tau) = m_1 - (V_2\ b - V_1\ a)\ (\tau - \tau_0)$, and the timer vector is $\boldsymbol{\nu}(\tau) = [\tau - \tau_0,\ 0,\ \tau - \tau_0,\ 0]^T$. ■

3.4 Admissible IFS vectors

We use linear inequalities to characterize the set of *all* admissible firing speed vectors $\mathcal{S}$. Each IFS vector $\boldsymbol{v} \in \mathcal{S}$ represents a particular mode of operation of the system described by the net. As discussed in detail in Subsection 4.1, the system operator may choose among all possible modes of operation, the best according to a given objective.

The set of admissible IFS vectors form a convex set described by linear equations.

Definition 3.10 (admissible IFS vectors) *Let $\langle N, \boldsymbol{m} \rangle$ be an FOHPN system with n_c continuous transitions and incidence matrix $\boldsymbol{C}$. Let*

- *$T_\mathcal{E}(\boldsymbol{m}) \subset T_c$ ($T_\mathcal{N}(\boldsymbol{m}) \subset T_c$) be the subset of continuous transitions enabled (not enabled) at $\boldsymbol{m}$,*
- *$P_\mathcal{E} = \{p \in P_c \mid m_p = 0\}$ be the subset of empty continuous places.*

Any admissible IFS vector $\boldsymbol{v} = [v_1, \cdots, v_{n_c}]^T$ *at $\boldsymbol{m}$ is a feasible solution of the following linear set:*

$$\begin{cases} (a)\ V_j - v_j \geq 0 & \forall t_j \in T_\mathcal{E}(\boldsymbol{m}) \\ (b)\ v_j - V'_j \geq 0 & \forall t_j \in T_\mathcal{E}(\boldsymbol{m}) \\ (c)\ v_j = 0 & \forall t_j \in T_\mathcal{N}(\boldsymbol{m}) \\ (d)\ \sum_{t_j \in T_\mathcal{E}} \boldsymbol{C}(p, t_j) \cdot v_j \geq 0 & \forall p \in P_\mathcal{E}(\boldsymbol{m}) \\ (e)\ v_j \geq 0 & \forall t_j \in T_c \end{cases} \tag{6}$$

A part from the non-negativity constraint (e) the total number of constraints that define this set is: $2\ card\{T_\mathcal{E}(\boldsymbol{m})\} + card\{T_\mathcal{N}(\boldsymbol{m})\} + card\{P_\mathcal{E}(\boldsymbol{m})\}$. *The set of all feasible solutions is denoted $\mathcal{S}(N, \boldsymbol{m})$.* ■

Constraints of the form (6.a), (6.b), and (6.c) follow from the firing rules of continuous transitions. Constraints of the form (6.d) follow from (4), because if a continuous place is empty then its fluid content cannot decrease. Note that if $V'_i = 0$, then the constraint of the form (6.b) associated to t_i reduces to a non–negativity constraint on v_i.

Example 3.11 Let $\langle N, \boldsymbol{m}(\tau_0) \rangle$ be the continuous net in Fig. 1.a. If $m_1 > 0$, according to the previous definition, the set $\mathcal{S}(N, \boldsymbol{m}(\tau_0))$ is defined by the following inequalities:

$$\begin{cases} V_1 - v_1 \geq 0 \\ V_2 - v_2 \geq 0 \\ v_1, v_2 \quad \geq 0 \end{cases} \tag{7}$$

If $m_1 = 0$, we add to the constraint set $\mathcal{S}(N, \boldsymbol{m}(\tau_0))$ the additional constraint $\{a\ v_1 - b\ v_2 \geq 0\}$ associated to the empty place p_1. ■

3.5 Main differences between the FOHPN model and the HPN of David and Alla

There are two main differences between our model and the one proposed in (Alla and David, 1998*b*).

The definition of continuous transitions enabling proposed in (Alla and David, 1998*b*) requires that a weakly enabled transition be "fed", i.e. there exists an upstream transition strongly enabled feeding it. According to this definition, two transitions in a cycle as depicted in Fig. 2.a are not enabled and the cycle is blocked, while according to our definition they are both weakly enabled and the cycle is not blocked. To overcome this limitation, David and Alla introduced the concept of ϵ-marking (see (David and Alla, 2005) for more details): if an arbitrary small marking is initially assigned to any of the two places of the cycle in Fig. 2.a, then both transitions can be considered weakly enabled. Thus, in this generalized framework it is possible to assign to empty cycles two semantics: blocked cycles (those that are empty) and non-blocked cycles (those ϵ-marked). We believe that blocked cycles are not a useful modeling feature for manufacturing systems of practical interest, thus we have chosen to keep just the second semantics. However, one may also adopt for FOHPNs the enabling definition used in (David and Alla, 2005).

Another difference with (Alla and David, 1998*b*) is that we have also introduced minimum firing speeds for continuous transitions. As a consequence of this, as shown in (Balduzzi *et al.*, 2000), the set $\mathcal{S}(N, \boldsymbol{m})$ defined by (6) may not admit feasible solutions in some cases. As an example, consider the net in Fig. 2.b, where transitions t_1 and t_2 have $(V'_1, V_1) = (0, 2)$ and $(V'_2, V_2) = (3, 5)$. If $m_1 = 0$ and place p_2 is marked, then there is no feasible solution to the constraint set:

$$\begin{cases} 0 \leq v_1 \leq 2 \\ 3 \leq v_2 \leq 5 \\ v_2 \leq v_1 \end{cases}$$

i.e. no admissible modes of operation is possible. This is a useful indication for the system designer that the system does not satisfy the requirements. Note that when place p_2 is not marked, transition t_2 is disabled, hence its IFS is $v_2 = 0$ and any $v_1 \in [0, 2]$ satisfies the constraint set, regardless of the value of m_1.

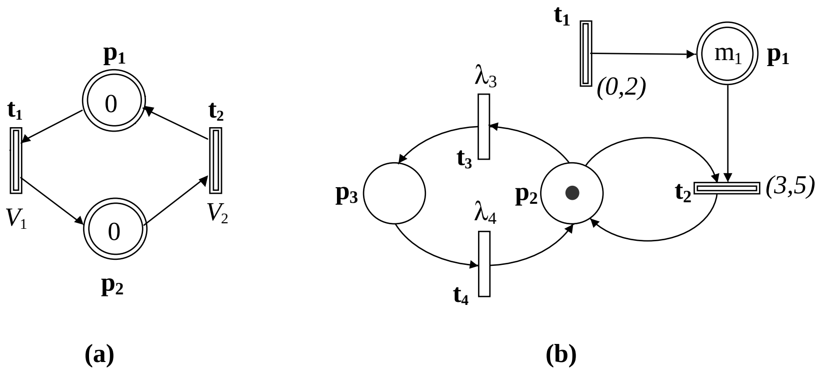

Fig. 2. (a) A FOHPN with an empty cycle. (b) A FOHPN which may have no admissible mode of operation.

4 Analysis and control of FOHPNs

Different analysis and control problems in the framework of FOHPNs have been investigated in the last years (Balduzzi *et al.*, 2000; Balduzzi *et al.*, 2001*a*), that are briefly summarized in this section.

4.1 Control

In the previous section we have shown how appropriate linear inequalities can be used to define the set of all admissible firing speed vectors $\mathcal{S}$. Each vector $v \in \mathcal{S}$ represents a particular mode of operation of the system described by the net, and among all possible modes of operation, the system operator may choose the best according to a given objective. Some examples are given in the following.

- **Maximize flows**. In an FOHPN we may consider as optimal the solution $\boldsymbol{v}^*$ of (6) that maximizes the performance index $J = \mathbf{1}^T \cdot \boldsymbol{v}$ which is of course intended to maximize the sum of all flow rates. In the manufacturing domain this may correspond to maximizing machines utilization.
- **Maximize outflows**. In an FOHPN we may want to maximize the performance index $J = \boldsymbol{c}^T \cdot \boldsymbol{v}$ where

$$c_j = \begin{cases} 1 \text{ if } t_j \text{ is an exogenous transition,} \\ 0 \text{ if } t_j \text{ is an endogenous transition.} \end{cases}$$

  In the manufacturing domain this may correspond to maximizing throughput.
- **Minimize stored fluid**. In an FOHPN we may want to minimize the derivative of the marking of a place $p \in P_c$. This can be done by minimizing

the performance index $J = \boldsymbol{c}^T \cdot \boldsymbol{v}$ where

$$c_j = \begin{cases} \boldsymbol{C}(p, t_j) & \text{if } t_j \in p^{(c)} \cup {}^{(c)}p, \\ 0 & \text{otherwise.} \end{cases}$$

In the manufacturing domain this may correspond to minimizing the work–in–process (WIP).

Note that this approach has several advantages with respect to other approaches proposed in the literature, e.g., (Dubois *et al.*, 1994), where an iterative algorithm was given to determine one admissible vector. In fact, we can explicitly define the set of all admissible IFS vectors in a given macro-state and not just compute a particular vector. Then, we compute a particular (optimal) IFS vector solving a Linear Programming Problem (LPP), rather than by means of an iterative algorithm, whose convergence properties may not be good. Finally, as discussed in the following subsection, linear programming leads to sensitivity analysis, which plays an essential role in performance evaluation and optimization.

However, the above control procedure still suffers of a serious drawback. In fact, the set $\mathcal{S}$ corresponds to a particular system macro-state. Thus, our optimization scheme can only be *myopic*, in the sense that it generates a piecewise optimal solution, i.e. a solution that is optimal only in a macro-period.

At present, we are looking for alternative solutions that are not myopic, but this is still an open issue. We believe that the approach used in (Bemporad *et al.*, 2004) to optimally control CPNs could be be successfully applied also in the case of FOHPNs, but we still have to verify this conjecture.

4.2 Reachability analysis

In (Balduzzi *et al.*, 2001*a*) we considered the untimed version of FOHPNs. Within this framework, we defined a particular class of nets, called *Single–Rate Hybrid Petri Nets* (SRHPNs). In a SRHPN the continuous evolution is such that the marking of all continuous places increases with a *single rate* that depends on the place, i.e., $\forall p_i \in P_c, \dot{m}_i = v_i$. Here in general $v_i \neq v_j$ for $i \neq j$, thus the activity function of a net in this class is equivalent to that of a timed automaton with skewed clocks (Alur *et al.*, 1993). In (Balduzzi *et al.*, 2001*a*) we proved that the reachability problem for this class can be reduced to the reachability problem of an equivalent discrete net and thus it is decidable.

Example 4.1 The FOHPN in Fig. 3 is a SRHPN. It represents a production system with two continuous flows of parts (type 1 and type 2) that are put into two buffers (places p_1 and p_2). The batch processing of parts, represented by the cycle of discrete transitions, requires first a unit of part type 1, then a unit of part type 2 and then again a unit of part type 1. ■

In (Balduzzi *et al.*, 2001*a*) we proved the following result.

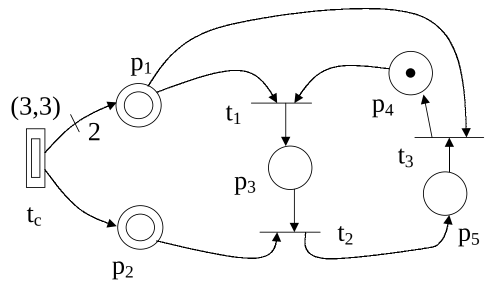

Fig. 3. A Single–Rate Hybrid Petri Net.

Corollary 4.2 *(Balduzzi* et al.*, 2001*a*)*The reachability problem is decidable for SRHPN.

In (Balduzzi *et al.*, 2001*a*) we also shown constructively how the reachability problem of a SRHPN can be reduced to that of a corresponding discrete Petri net.

The goal of our future research in this topic consists in investigating if such result can be extended to more general classes of FOHPNs, and eventually be disproved for the most general model.

5 Modelling and simulation of distributed manufacturing systems

This section shows the efficiency of FOHPN in the modeling and control of large and complex systems such as *Distributed Manufacturing Systems* (DMS).

5.1 The DMS system description

The DMS structure is typically described by a set of facilities with materials that flow from the sources of raw materials to manufacturers and onwards to assemblers and consumers of finished products. DMS facilities are connected by transporters of materials, semi-finished goods and finished products. More precisely, the DSM entities can be summarized as follows.

(1) *Suppliers:* a supplier is a facility that provides raw materials, components and semi-finished products to manufacturers that make use of them.
(2) *Manufacturers and assemblers:* manufacturers and assemblers are facilities that transform input raw materials/components into desired output products.
(3) *Logistics and transporters:* storage systems and transporters play a critical role in distributed manufacturing. The attributes of logistics facilities are storage and handling capacities, transportation times, operation and inventory costs.
(4) *Retailers or customers:* retailers or customers are sink nodes of material flows.

Here, some parts of the logistics, such as storage buffers, are considered pertaining to manufacturers, suppliers and customers. Moreover, transporters connect the different stages of the production process.

The dynamics of the distributed production system is traced by the flow of products between facilities and transporters. Because of the large amount of material flowing in the system, we model a DMS as a hybrid system: the continuous dynamics models the flow of products in the DMS, the manufacturing and the assembling of different products and its storage in appropriate buffers. Hence, the levels of buffers accommodating products are represented by continuous states describing the amount of fluid material that the resources store. Moreover, we also consider discrete events occurring stochastically in the system, such as:

(a) the blocking of the raw material supply, e.g. modeling the occurrence of labor strikes, accidents or stops due to the shifts;
(b) the blocking of transport operations due to the shifts or to unpredictable events such as jamming of transportation routes, accidents, strikes of transporters etc.;
(c) the beginning and the end of a request from a DMS facility.

5.2 A modular DMS model based on FOHPN

This section proposes a modular approach using FOHPN to model DMS based on the idea of the *bottom-up approach* (Zhou and Venkatesh, 1998). Such a method can be summarized in two main steps: *decomposition* and *composition*. Decomposition involves dividing the system into several subsystems. In DMS this division can be performed based on the determination of distributed system facilities (i.e., suppliers, manufacturers, transporters and customers). All these subsystems are then modeled by FOHPN. Finally, composition involves the interacting of these sub-models into a complete model, representing the whole DMS.

In the following we present the main FOHPN models of the elementary subsystems composing a DMS.

5.2.1 The supplier module

The supplier is modeled as a continuous transition and two continuous places (see places p_S, p'_S and transition t_S in Fig. 4.a). The continuous transition t_S models the arrival of raw material in the system at a bounded rate v_S that belongs to the interval $\mathcal{C}(t_S) = [V_{S,min}, V_{S,max}]$. We consider the possibility that the supply of raw material is blocked for a certain time period. This situation is modeled by two exponentially distributed transitions (t_k and t'_k) and two discrete places (p_k and p'_k). In particular, place p_k represents the operative state of the supplier, and p'_k is the non-operative state (see Fig. 4.a). The blocking and the restoration of the raw material supply corresponds to the firing of transitions t_k and t'_k, respectively.

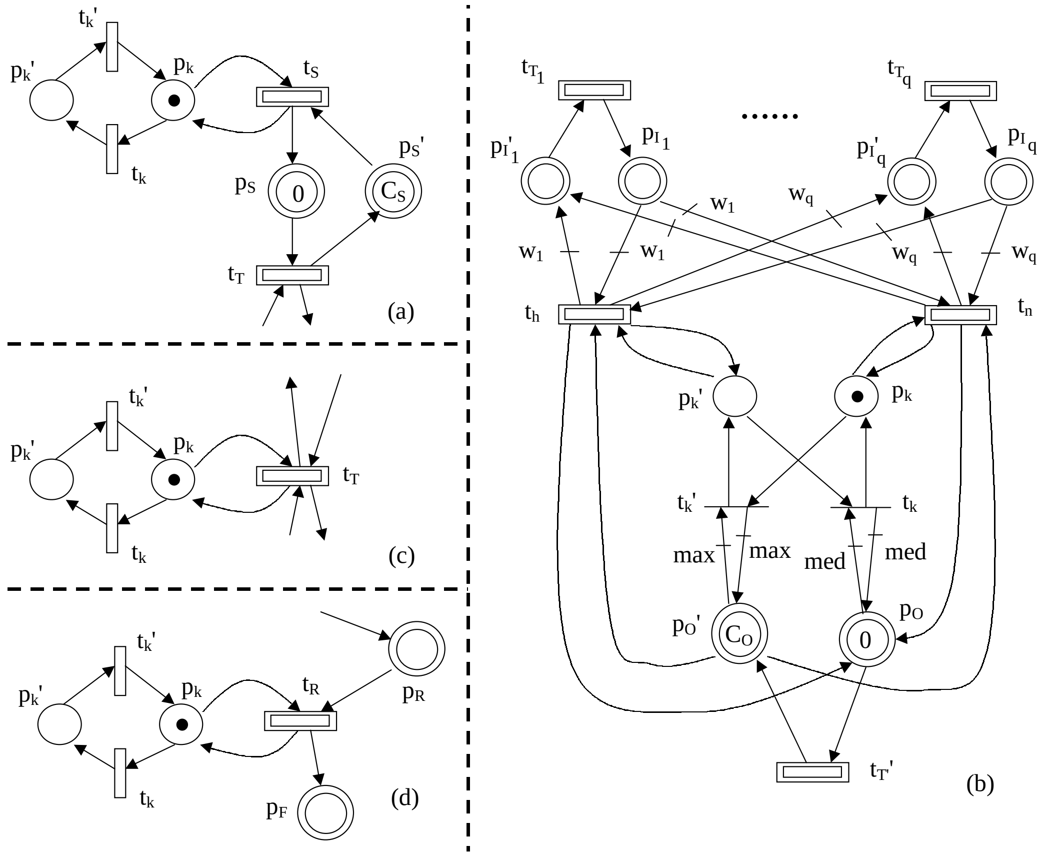

Fig. 4. The FOHPN models of a supplier (a); a manufacturer or assembler (b); a logistics (c), and a transporter (d).

The continuous place p_S models the raw material buffer of finite capacity C_S, and p'_S represents the corresponding available capacity. Thus, at each time instant, with no ambiguity in the notation, we can write $m_S + m'_S = C_S$.

Note that for sake of clarity, in Fig. 4.a we have also reported the transition t_T that, as discussed later (see Subsection 5.2.3), models the transport operation that corresponds to the withdrawal of material from the buffer.

5.2.2 The manufacturer and assembler module

Manufacturers and assemblers processing q types of semi-finished products (or components) to obtain a final product can be modeled by the FOHPN in Fig.4.b. Two places and one transition are associated to each product type r, for $r = 1, \ldots, q$. More precisely, place p_{I_r} is the input buffer of finite capacity C_{I_r} storing the input goods of type r. The corresponding place p'_{I_r} models the available buffer space so that $m_{I_r} + m'_{I_r} = C_{I_r}$. Moreover, each continuous transition t_{T_r} represents the transport of products of type r.

The output buffer of capacity C_O storing the output product is modeled by place p_O representing the occupied buffer level and by place p'_O modeling the corresponding available capacity. Thus at any time instant $m_O + m'_O = C_O$.

We assume that the production rate is a function of the level of the output buffer. In particular, we assume that the production rate of the facility changes as a function of the available space in the output buffer. Indeed, we consider two different ranges of rate for the production: a nominal range $\mathcal{C}(t_n) = [0, V_n]$ associated to the continuous transition t_n and a high range $\mathcal{C}(t_h) = [0, Vh]$, with $V_h > V_n$, associated to the continuous transition t_h. If $m'_0(t) < \max$, where max is value that is established a priori (i.e., the inventory level is large enough) the manufacturer works at a rate within the nominal range. When the buffer level decreases (i.e., the available buffer free is equal to $m'_O(t) = \max$), the immediate transition t'_k fires. After its firing $m'_k = 1$ so the facility may work at a higher rate in $\mathcal{C}(t_h)$ and the output buffer is replenished more rapidly. Analogously, if the level of the buffer reaches an intermediate value $m_O(t) = \text{med}$, then transition t_k is enabled and can fire, leading the facility rate at a value in the nominal range.

Note that, since an assembly operation can be performed by assembling different quantities of each product, different weights have been assigned to the arcs from p_{I_r} to t_h and t_n and from t_h and t_n to p_{I_r}, for $r = 1, \ldots, q$.

Finally, transition t'_T depicts the transport facility connecting the output buffer with the other system facilities.

5.2.3 The logistics module

The FOHPN model of logistics is reported in Fig. 4.c. The transporter connecting the different facilities is modeled by a continuous transition t_T that describes the flow of material from a facility to a subsequent one at a bounded rate $v_T \in [0, V_T]$. Moreover, the random stop of the material transport is represented by two places $p_k, p'_k \in P_d$ and two exponentially distributed transitions t_k and t'_k. If place p_k is marked the transport is operative; if p'_k is marked the transportation is no longer enabled.

Note that this model is rather simplified because the delay on transportation is not taken into account. However, it can be easily considered using appropriate primitives such as in (Corriga *et al.*, 1997) or (Júlvez and Boel, 2005).

5.2.4 The retailer module

The FOHPN model of a retailer is reported in Fig. 4.d. It is constituted by a continuous place p_R modeling a buffer (for simplicity we consider an infinite capacity buffer) associated with a final product of a certain type. The continuous transition t_R models the acquisition of the final product by the retailer. The continuous place p_F models the system output and collects all the products requested by the retailer.

Note that in general place p_F has more than one input transition, each one modeling the acquisition of a different product type. The input arcs have in general different weights, e.g., depending on the volume of the corresponding product type.

Finally, since we assume that the request of the products is managed by discrete random events (i.e., by a random demand), as in the previous elementary modules, we introduce the cycle given by two discrete timed transitions (t_k and t'_k) and two discrete places (p_k and p'_k).

5.2.5 Remarks

In all the above models we assumed that the discrete state of the system only depends on the firing of some exponentially distributed transitions. Such an assumption can be easily removed and more general cases may be considered, e.g., periodic behaviours in the demand. We do not handle these cases here for sake of brevity and because we already addressed similar issues in (Furcas *et al.*, 2001; Balduzzi *et al.*, 2001*b*) when considering inventory and manufacturing systems.

Moreover, here we have also omitted the description of the financial transactions and information flows occurring in the DMS. As discussed in (Furcas *et al.*, 2001) in the case of inventory management, such an information can be straightforwardly included in the model adding appropriate FOHPN modules.

5.3 An application example of DMS

To illustrate the modeling technique, we consider the DMS depicted in Fig. 5 composed by two suppliers $S1$ and $S2$, two subassembly manufacturers $M1$ and $M2$, two assemblers $A1$ and $A2$, and two retailers $R1$ and $R2$. Moreover, six logistics service providers, $T1$ to $T6$, suitably connect the DMS facilities that are located in different geographical sites. We assume that the system produces two product brands E and F, ordered by both buyers. Such products are obtained by two assemblers that assemble (in the same proportions) two types of products (C and D) obtained from two manufacturers. The subassemblies C and D are in turn produced by the manufacturers, which receives the components of type A and B by the suppliers. Moreover, we assume that the DMS is managed by the *Make To Stock* (MTS) policy. This means that the system is managed by a push strategy, so that end customers are satisfied from stock of inventory of finished goods.

The whole system is modeled merging all the elementary modules described in the previous section. The resulting FOHPN is reported in Fig. 6 where each facility module is depicted within dashed boxes.

5.3.1 Simulation and optimization

The DMS dynamics is analyzed via numerical simulation using the data reported in Table 1, where we can read the manufacturer and assembler production ranges of rate, the range of the transportation speeds, and the average firing delays of discrete stochastic transitions. Table 2 shows further data necessary to completely describe the system, namely the initial marking of continuous places, the buffer capacities for the inventories of each stage, and the value of the non-unitary edge weights. The initial marking of discrete places is shown in Fig. 6.

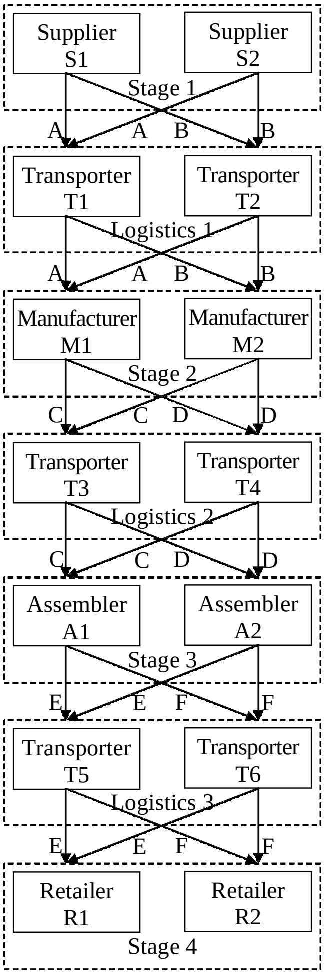

Fig. 5. The DMS considered in Subsection 5.3.

In order to analyze the DMS behavior, some basic performance indices can be considered.

(i) The throughput T_i associated to each retailer Ri, $i = 1, 2$, i.e., the average number of products obtained by each buyer in a time unit.
(ii) The system throughput $T = T_1 + T_2$.
(iii) The average input inventory in manufacturer Mi with $i = 1, 2$ (I_{Mi}, $i = 1, 2$) and in assembler Ai with $i = 1, 2$ (I_{Ai}, $i = 1, 2$) during the run time TP.
(iv) The average output inventory in manufacturer Mi with $i = 1, 2$ (O_{Mi}, $i = 1, 2$) and in assembler Ai with $i = 1, 2$ (O_{Ai}, $i = 1, 2$) during the run time TP.
(v) The average system inventory SI, i.e., the sum of the amount of product storage in all buffers during the run time TP.
(vi) The lead time $LT = SI/T$ that is a measure of the time spent by the DMS to convert the raw material in final product (Viswanadham, 2000).

The FOHPN model has been implemented and simulated using Matlab. Such a matrix-based software appears particularly appropriate for simulating the

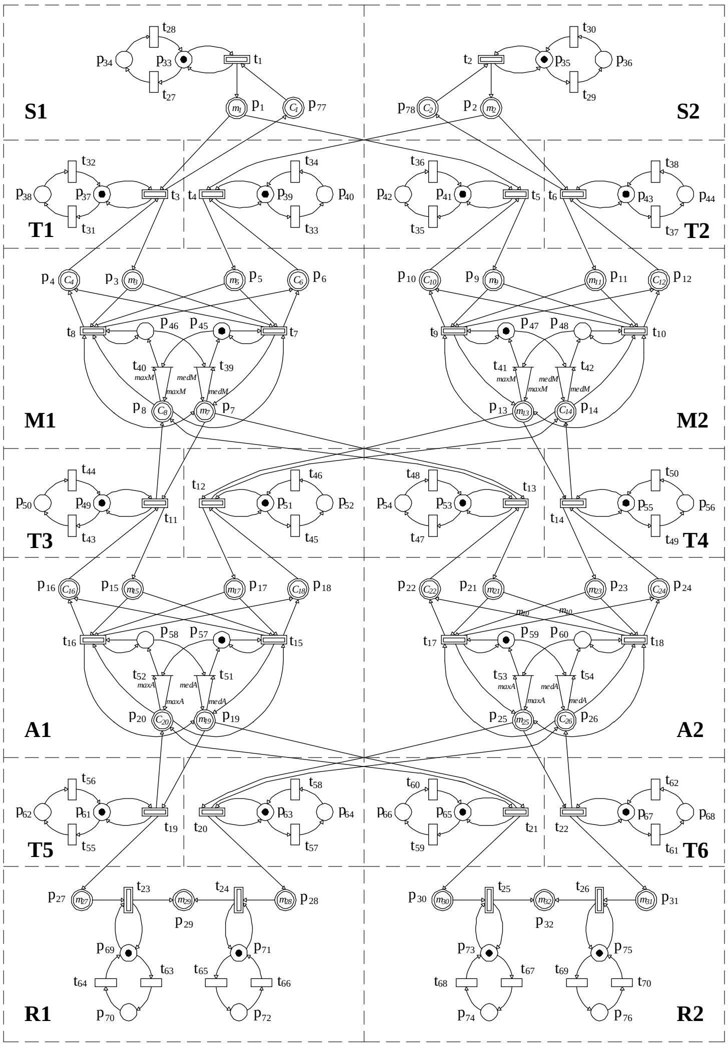

Fig. 6. The FHPN model of the DMS in Fig. 5.

FOHPN dynamics based on the matrix formulation of the marking update described in Section 3. In particular, the software we have developed is able to integrate modeling and simulation of hybrid systems with the solution of constrained optimization problems, i.e., the IFS vector choice within the set of admissible values by optimizing a particular objective function.

More in detail, after defining the system parameters and the initial marking, the main simulation program first selects the value of each transition timer set point, then determines the set of IFS admissible vectors and solves the optimization program by a suitable Matlab routine; subsequently determines the next macro-event to occur using an appropriate routine that singles out the enabled transitions. Hence the simulation determines the next marking with the matrix formulation of the marking update described in Section 3, and finally updates the set point of all transitions so that the next macro-period may be simulated.

Continuous transitions	$[V_{min}, V_{max}]$ (Units per hours)	Discrete Transitions	Average firing delay (hours)
t_1	[0, 8]	t_{27} t_{29}	10
t_2	[0, 10]	t_{28} t_{30}	14
t_3 t_6 t_{14} t_{22}	[0, 10]	t_{31} t_{37} t_{47}	9
t_4 t_7 t_{13} t_{17} t_{19}	[0,8]	t_{49} t_{55} t_{57}	9
t_5 t_{12} t_{20}	[0,9]	t_{32} t_{38} t_{48}	15
t_8	[8,20]	t_{50} t_{56} t_{58}	15
t_9	[0, 12]	t_{33} t_{35} t_{43}	10
t_{10}	[12,30]	t_{45} t_{59} t_{61}	10
t_{11} t_{21}	[0, 11]	t_{34} t_{36} t_{44}	14
t_{15}	[0, 6]	t_{46} t_{60} t_{62}	14
t_{16}	[6, 15]	t_{63} t_{65}	10
t_{17}	[0, 8]	t_{64} t_{66}	14
t_{18}	[8, 20]	t_{67} t_{69}	10
t_{23}	[0, 3]	t_{68} t_{70}	14
t_{24} t_{25}	[0, 4]		
t_{26}	[0, 5]		

Table 1
Firing speeds of continuous transitions and average firing delay of discrete transitions.

Initial marking (product units)	Capacities (product units)
m_1, m_2=20	C_1, C_2=200
m_3, m_5, m_9, m_{11}=20	C_4, C_6, C_{10}, C_{12}=100
m_7, m_{13}=20	C_8, C_{14}=80
m_{15}, m_{21}, m_{17}, m_{23}=20	C_{16}, C_{22}, C_{18}, C_{24}=100
m_{19}, m_{25}=20	C_{20}, C_{26}=80
m_{27}, m_{28}, m_{29}=0	
m_{30}, m_{31}, m_{32}=0	
Edge weights (product units)	
max_M=190	max_A=190
med_M=50	med_A=50

Table 2
Initial marking of continuous places, capacities and edge weights.

All the indices have been estimated by simulation runs of a time period $TP = 480$ hours (the hour is the considered time unit), and 1000 independent replications. Three different operative conditions, denoted OCi with $i = 1, 2, 3$, have been taken into account, each one corresponding to a different criterion in the choice of the IFS vector within the set of admissible values.

- *First Operative Condition (OC1).* At each macro-period the IFS vector $\boldsymbol{v}$ is taken equal to the intermediate value in the interval of admissible speed values.
- *Second Operative Condition (OC2).* At each macro-period the IFS vector $\boldsymbol{v}$ is selected so as to maximize the sum of all flow rates (see first item in Subsection 4.1).
- *Third Operative Condition (OC3).* At each macro-period the IFS vector $\boldsymbol{v}$ is selected so as to minimize the stored volume (see third item in Subsec-

Performance index	*OC1*	*OC2*	*OC3*
T_1 (product units /hours)	0.59	1.08	0.57
T_2 (product units /hours)	0.71	1.33	0.71
T (product units /hours)	1.30	2.41	1.29
LT (hours)	209.04	196.85	125.87
SI (product units)	268.51	469.36	159.45

Table 3
Performance indices obtained under the three operative conditions.

tion 4.1).

The main results of the numerical simulations we carried out are summarized in Fig. 7 to 9. In particular, Fig. 7 clearly shows that $OC2$ provides the best performances in terms of system throughput. This result is not surprising because in such operational condition the goal was exactly that of maximizing the sum of all flow rates. We can also observe that in this respect $OC1$ and $OC3$ give similar results.

Fig. 8 shows the average inventory in manufacturers and assemblers in all the operative conditions. The figure shows that, also thanks to the dependence of the manufacturers and assemblers production rates with their output inventories, the DMS is able to keep stocks at a satisfactorily high level, so that the demand is satisfied and the inventory is not excessive. In particular, as expected, $OC2$ provides the highest inventories and $OC3$ the lowest stocks, while $OC1$ provides intermediate values.

Finally, Fig. 9 shows the value of the lead time in the different considered cases. In particular, we may observe that the highest value of LT is obtained in the case of $OC1$, while the lowest value corresponds to $OC3$.

Summing up, we conclude that $OC2$ exhibits the highest throughput value but also high values of the system inventory and of the lead time; $OC3$ provides the lowest inventory levels and also the lowest values of LT. Finally, $OC1$ leads to the same throughput of $OC3$ but with higher values of the inventory levels and of the lead time, thus it is not an appropriate choice.

6 Conclusions

In this paper we considered a particular hybrid formalism, namely HPNs, that has been obtained as an extension (via fluidification) of the Petri net model for discrete event systems. The main contribution of this paper can be summarized as follows.

First, we provided a detailed discussion of the main HPNs models that have been presented in the literature.

Second, we focused our attention on a particular class, the First-Order Hybrid Petri nets. We presented it in detail, mentioning the main theoretical results

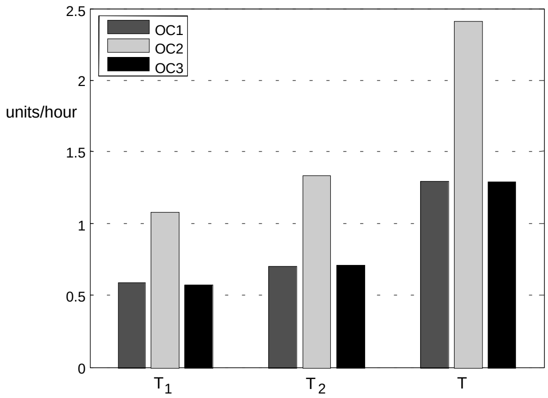

Fig. 7. Throughput under the different operative conditions.

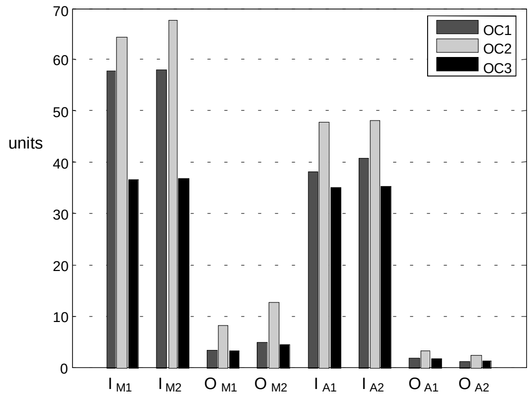

Fig. 8. Average inventory in manufacturers and in assemblers under the different operative conditions.

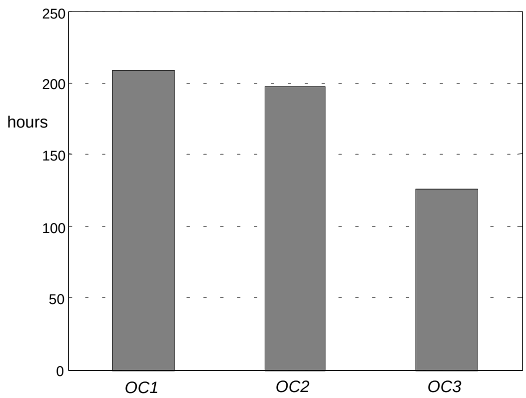

Fig. 9. Lead times under the different operative conditions.

that have been solved within this framework, and also mentioning the main open issues.

Finally, we showed how FOHPNs can be efficiently used to model DMS, and how interesting optimization problems can be solved via numerical simulation, by simply solving on-line a certain number of LPPs.

Acknowledgments

The authors would like to thank Aldo Piccaluga and Daniele Corona who contribute to create and maintain the webpage "Bibliography on Hybrid Petri nets" (Giua, 2006), that has been the starting point for the development of Section 2.

The authors also want to thank Agostino M. Mangini who collaborated in the development of the Matlab programs that have been used for the simulation of the DMS in Section 5.

References

Ait-Yahia, A., N. Zerhouni, A. Elmoudni and M. Ferney (1995). On the modeling and simulation of variable speed continuous Petri nets by design/CPN. In: *Proc. 1995 EUROSIM Conf., EUROSIM '95*. Vienna, Austria.

Alla, H. (1995). Modelling and simulation of event driven systems by Petri nets. In: *Proc. ADEDOPS Workshop*. London, UK.

Alla, H. and R. David (1988). Modelling of production systems by continuous Petri nets. In: *Proc. 3rd Int. Conf. on CAD/CAM Robotics and Factories of the Future*. Southfield, MI, USA.

Alla, H. and R. David (1998*a*). Continuous and hybrid Petri nets. *Journal of Circuits Systems Computers* **8**(1), 159–88.

Alla, H. and R. David (1998*b*). A modeling and analysis tool for discrete events systems: continuous Petri net. *Performance Evaluation* **33**(3), 175–199.

Allam, M. and H. Alla (1997). Modelling production systems by hybrid automata and hybrid Petri nets. Proc. IFAC-IFIP-IMACS Conf. on Control of Industrial Systems.

Allam, M. and H. Alla (1998). Modeling and simulation of an electronic component manufacturing system using hybrid Petri nets. *IEEE Trans. on Semiconductor Manufacturing* **11**(3), 374–383.

Alur, R., C. Courcoubetis, T.A. Henzinger and P.H. Ho (1993). Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. Vol. 736 of *Lecture Notes in Computer Science*. Springer Verlag.

Amer-Yahia, C., N. Zerhouni, A. El-Moudni and M. Ferney (1995). State equation and stability of a class of continuous Petri nets. application to manufacturing lines. Proc. ETFA'95 Symp. on Emerging Technologies and Factory Automation. Paris, France.

Amer-Yahia, C., N. Zerhouni, A. El-Moudni and M. Ferney (1996). State variable description and controllability of a class of continuous Petri nets. Proc. 1996 IEEE Int. Symp. On Circuits and Systems. Atlanta, GA, USA.

Amer-Yahia, C., N. Zerhouni, M. Ferney and A. El Moudni (1997). Modelling of biological systems by continuous Petri net. Proc. 3rd IFAC Symp. Modelling and Control in Biomedical Systems. Warwick, UK.

Amrah, A., N. Zerhouni and A. El Moudni (1996). Constrained state feedback control of a class of discrete event systems modelled by continuous Petri nets. Proc. 4th Int. Conf. on Control, Automation, Robotics and Vision. Singapore.

Amrah, A., N. Zerhouni and A. El Moudni (1997*a*). Control of discrete event systems modeled by continuous Petri nets: case of opened assembly manufacturing lines. Proc. ECC'97, European Control Conf.. Brussels, Belgium.

Amrah, A., N. Zerhouni and A. El Moudni (1997*b*). Control of discrete event systems modeled by continuous Petri nets: case of opened manufacturing lines. Proc. 1997 IEEE Int. Conf. on Robotics and Automation. Albuquerque, NM, USA.

Amrah, A., N. Zerhouni and A. El Moudni (1998*a*). Flow control of manufacturing systems. case of opened manufacturing lines. Proc. CESA'98: Computational Engineering in Systems Application. Nabeul-Hammamet, Tunisia.

Amrah, A., N. Zerhouni and A. El Moudni (1998*b*). On the control of manufacturing lines modelled by controlled continuous Petri nets. *Int. Journal of Systems Science* **29**(2), 127–137.

Andreu, D., J.C. Pascal and R. Valette (1995). Interaction of discrete and continuous parts of a batch process control system. In: *Proc. Workshop on Analysis and Design of Event-Driven Operations in Process Systems*. London, UK.

Andreu, D., J.C. Pascal and R. Valette (1996). Events as a key of a batch process control system. In: *Proc. CESA'96, Symp. On Discrete Events and Manufacturing Systems*. Lille, France.

Andreu, D., J.C. Pascal, H. Pingaud and R. Valette (n.d.). Batch process modeling using Petri nets. In: *Proc. 1994 IEEE Int. Conf. on Systems, Man, and Cybernetics*.

Antsaklis, P.J. and X.D. Koutsoukos (1998). On hybrid control of complex systems: a survey. In: *Proc. 3rd Int. Conf. on Automation of Mixed Processes*. Reims, France.

Audry, N. and F. Prunet (1995). Controlled batches Petri nets. In: *Proc. 1995 IEEE Int. Conf. on Systems, Man and Cybernetics, Intelligent Systems*. Vancouver, BC, Canada.

Audry, N., I. Demogodin and F. Prunet (1994). Modelling of high throughput production lines by using generic models described in batches Petri nets. In: *Proc. 1994 IEEE Int. Conf. on Robotics and Automation*.

Bail, J. Le, H. Alla and R. David (1992). Asymptotic continuous Petri nets: an efficient approximation of discrete event systems. Proc. 1992 IEEE Int. Conf. on Robotics and Automation. Nice, France.

Bail, J. Le, H. Alla and R. David (1993). Asymptotic continuous Petri nets. *Discrete Event Dyn. Syst., Theory and Applications* **2**(3-4), 235–263.

Balduzzi, F., A. Di Febbraro, A. Giua and C. Seatzu (2001*a*). Decidability results in first-order hybrid petri nets. *J. of Discrete Event Dynamic Systems: Theory and Applications.*

Balduzzi, F., A. Giua and C. Seatzu (2001*b*). Modeling and simulation of manufacturing systems with first–order hybrid Petri nets. *Int. J. of Production Research* **39**, 285–282.

Balduzzi, F., A. Giua and G. Menga (2000). First–order hybrid Petri nets: a model for optimization and control. *IEEE Trans. Robotics and Automation* **16**, 382–399.

Bemporad, A., J. Júlvez, L. Recalde and M. Silva (2004). Event-driven optimal control of continuous Petri nets. Proc. 43th IEEE Conf. on Decision and Control. Atlantis, Paradise Island, Bahamas.

Bemporad, A., M. Morari, V. Dua and E.N. Pistikopoulos (2002). The explicit linear quadratic regulator for constrained systems. *Automatica* **38**(1), 3–20.

Champagnat, R., P. Esteban, H. Pingaud and R. Valette (1998). Modeling and simulation of a hybrid system through PR/TR PN-DAE model. In: *Proc. 3rd Int. Conf. on Automation of Mixed Processes.* Reims, France.

Champagnat, R., R. Valette, J.C. Hochon and H. Pingaud (2001). Modeling, simulation and analysis of batch production systems. *J. of Discrete Event Dynamic Systems: Theory and Applications* **11**, 119–136.

Chen, H. and H.-M. Hanisch (1998). Hybrid net condition/event systems for modeling and analysis of batch processes. Proc. 3rd Int. Conf. on Automation of Mixed Processes. Reims, France.

Ciarlo, G., D.M. Nicol and K.S. Trivedi (1999). Discrete-event simulation of fluid stochastic Petri nets. *IEEE Trans. on Software Engineering* **2**(25), 207–217.

Colom, J.M. and M. Silva (1990). Improving the linearly based characterization of P/T nets. Vol. 483 of *Lecture Notes in Computer Science*. Springer Verlag.

Corriga, G., A. Giua and G. Usai (1997). Petri net modeling of irrigation canal networks. In: *Proc. Int. Work. on Regulation of Irrigation Canals.* Marrakech, Morocco.

Daubas, B., A. Pag‘es and H. Pingaud (1994). Combined simulation of hybrid processes. In: *Proc. 1994 IEEE Int. Conf. on Systems, Man, and Cybernetics.*

David, R. (1997). Modeling of hybrid systems using continuous and hybrid Petri nets. In: *Proc. Conf. on Petri Nets and Performances Evaluation.* Saint Malo, France.

David, R. and H. Alla (1987). Continuous Petri nets. In: *Proc. 8th European Workshop on Application and Theory of Petri Nets.* Zaragoza, Spain.

David, R. and H. Alla (1993). Autonomous and timed continuous Petri nets. Proc. 11th Int. Conf. on Application and Theory of Petri Nets. Paris, France.

David, R. and H. Alla (2005). *Discrete, continous and hybrid Petri nets.* Springer, Berlin, Heidelberg.

Demongodin, I. (1999). Extended structures of batches Petri nets. In: *Proc. IEEE Int. Conf. on Systems Man & Cybernetics*. Tokyo, Japan.

Demongodin, I. (2001). Generalized batches Petri: hybrid model for high speed systems with variable delays. *J. of Discrete Event Dynamic Systems: Theory and Applications* **11**, 137–162.

Demongodin, I. and A. Giua (2002). Some analysis methods for continuous and hybrid petri nets. In: *Proc. IFAC World Congress*. Barcelona, Spain.

Demongodin, I. and N.T. Koussoulas (1998). Differential Petri nets: Representing continuous systems in a discrete-event world. *IEEE Trans. Automatic Control*.

Demongodin, I., M. Caradec and F. Prunet (1998). Fundamental concepts of analysis in batches Petri nets. In: *Proc. 1998 IEEE Int. Conf. on Systems, Man, and Cybernetics*. San Diego, CA,USA.

Demongodin, I., N. Audry and F. Prunet (1993). Batches Petri nets. In: *Proc. 1993 Int. Conf. on Systems, Man and Cybernetics, Systems Engineering in the Service of Humans*. Le Touquet, France.

Desrochers, A., T. J. Deal and M.P. Fanti (2005). Complex-valued token Petri nets. *IEEE Trans. on Automation Science and Engineering*.

Dotoli, M., M.P. Fanti and A.M. Mangini (2006*a*). Modelling distributed manufacturing systems via first order hybrid Petri nets. In: *Proc. 2nd IFAC Conference on Analysis and Design of Hybrid Systems*. Alghero, Italy.

Dotoli, M., M.P. Fanti, C. Meloni and M.C. Zhou (2006*b*). Design and optimization of integrated e-supply chain for agile and environmentally conscious manufacturing. *IEEE Trans. on Systems Man and Cybernetics - part A* **36**(1), 62–75.

Dubois, E. and H. Alla (1993). Hybrid Petri nets with a stochastic discrete part. Proc. 2nd European Control Conf.. Groningen, Netherlands.

Dubois, E., H. Alla and R. David (1994). Continuous Petri net with maximal speeds depending on time. Proc. 15th Int. Conf. on Application and Theory of Petri Nets. Zaragoza, Spain.

El-Fouly, T., N. Zerhouni, M. Ferney and A. El Moudni (1998). Modeling and analysis of manufacturing systems using a hybrid approach. Proc. 3rd Int. Conf. on Automation of Mixed Processes. Reims, France.

Flaus, J.-M. (1996). Hybrid flow nets for batch process modeling. In: *Proc. CESA'96, Symp. On Discrete Events and Manufacturing Systems*. Lille, France.

Flaus, J.-M. (1997). Hybrid flow nets for batch process modeling and simulation. In: *Proc. 2nd IMACS Symp. On Mathematical Modeling*. Vienna, Austria.

Flaus, J.-M. and G. Ollagnon (1997). Hybrid flow nets for hybrid processes modeling and control. In: *Proc. Int. Workshop on Hybrid and Real-Time Systems*. Grenoble, France.

Flaus, J.-M. and H. Alla (1997). Structural analysis of hybrid systems modelled by hybrid flow nets. In: *Proc. European Control Conference*. Brussels, Belgium.

Furcas, R., A. Giua, A. Piccaluga and C. Seatzu (2001). Hybrid Petri net modeling of inventory management systems. *European Journal of Automation, JESA* **35**, 417–434.

Gaujal, B. and A. Giua (2004). Optimal stationary behavior for a class of timed continuous Petri nets. *Automatica* **40**(9), 1505–1516.

Genrich, H.J. and I. Schuart (1998). Modeling and verification of hybrid systems using hierarchical coloured Petri nets. Proc. 3rd Int. Conf. on Automation of Mixed Processes. Reims, France.

Giua, A. (2006). http://www.diee.unica.it/ giua/bibliohpn.html.

Giua, A. and E. Usai (1998). Modeling hybrid systems by high-level Petri nets. *European Journal of Automation APII-JESA*.

Giua, A., M.T. Pilloni and C. Seatzu (2001). Modelling and simulation of a bottling plant using hybrid Petri nets. *Int. J. of Production Research* **43**, 1375–1395.

Gribaudo, M., M. Sereno, A. Horvàt and A. Bobbio (2001). Fluid stochastic Petri nets augmented with flush-out arcs: modelling and analysis. *J. of Discrete Event Dynamic Systems: Theory and Applications* **11**, 97–117.

Horton, G., V.G. Kulkani, D.M. Nicol and K.S. Trivedi (1998). Fluid stochastic Petri nets: theory, application, and solution techniques. *European Journal of Operation Research* **701**(1), 184–201.

Júlvez, J. and R. Boel (2005). Modelling and controlling traffic behaviour with continuous Petri nets. In: *Proc. 16th IFAC World Congress*. Prague, Czech Republic.

Júlvez, J., E. Jimènez, L. Recalde and M. Silva (2004). Design of observers for timed CPN systems. Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics. The Hague, The Netherlands.

Júlvez, J., L. Recalde and M. Silva (2002). On deadlock-freeness analysis of autonomous and timed continuous mono-T semiflow nets. Proc. 41th IEEE Conf. on Decision and Control. Las Vegas, USA.

Júlvez, J., L. Recalde and M. Silva (2003). On reachability in autonomous continuous Petri net systems. Vol. 2679 of *Lecture Notes in Computer Science*. Springer Verlag.

Komenda, J., N. Zerhouni and A. El Moudni (1998). Analysis of hybrid Petri nets based on the hybrid state equation. Proc. 1998 IEEE Int. Conf. on Systems, Man, and Cybernetics. San Diego, CA,USA.

Lefebvre, D. (1999). Feedback control designs for manufacturing systems modeled by continuous Petri nets.. *International Journal of Systems Sciences* **30**(6), 591–600.

Lefebvre, D. (2000*a*). Estimation of the production frequencies for manufacturing systems. *IMA J. of Management Mathematics*.

Lefebvre, D. (2000*b*). Firing speeds estimation for continuous Petri nets. Proc. 39th IEEE Conf. on Decision and Control. Sydney, Australia.

Mahulea, C., A. Giua, L. Recalde, C. Seatzu and M. Silva (2006*a*). Optimal control of timed continuous Petri nets via explicit mpc. Lecture Notes in Computer Science. Springer Verlag.

Mahulea, C., A. Ramirez Trevi noand L. Recalde and M. Silva (2006*b*). Decidable hybrid systems. *IEEE Trans. on Automation Science and Engineering.*

Mahulea, C., L. Recalde and M. Silva (2006*c*). On performance monotonicity and basic servers semantics of continuous petri nets. Michigan, USA.

Marsan, M. Ajmone, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis (1995*a*). *Modelling with Generalized Stochastic Petri Nets*. John Wiley & sons.

Marsan, M. Ajmone, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis (1995*b*). *Modelling with Generalized Stochastic Petri Nets*. Wiley Series in Parallel Computing, John Wiley and sons.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings IEEE* **77**, 541–580.

Nicol, D. M. and A.S. Miner (1995). The fluid stochastic Petri net simulator. Proc. 6th Int. Workshop on Petri Nets and Performance Models. Durham, NC, USA.

Petri, C.A. (1962). Kommunikation mit Automaten (Communication with Automata). PhD thesis.

Puri, A. and P. Varaiya (1996). Decidable hybrid systems. *Computer and Mathematical Modeling* **23**(11-12), 191–202.

Silva, M. and L. Recalde (2004). On fluidification of Petri net models: from discrete to hybrid and continuous models. *Annual Reviews in Control* **28**(2), 253–266.

Trivedi, K.S. and V. G. Kulkarni (1993). FSPNs: Fluid stochastic Petri nets. Vol. 691 of *Lecture Notes in Computer Science, M.Ajmone Marsan (Ed.)*. Springer Verlag.

Valentin-Roubinet, C. (1994*a*). Modeling and analysis methods for a class of hybrid dynamic systems. In: *Proc. Symp. ADPM'94*. Brussels, Belgium.

Valentin-Roubinet, C. (1994*b*). Modelling and simulation of sequential hybrid dynamic systems. In: *Proc. Workshop on Analysis and Design of Event-Driven Operations in Process Systems*. London, UK.

Valentin-Roubinet, C. (1998). Modeling of hybrid systems: Dae supervised by Petri nets. the example of a gas storage. In: *Proc. 3rd Int. Conf. on Automation of Mixed Processes*. Reims, France.

Vibert, D., C. Valentin-Roubinet and E. Niel (1997). A modelling method to take into account fluctuations of continuous variables in a class of hybrid systems. In: *European Control Conference*. Brussels, Belgium.

Viswanadham, N. (2000). *Analysis of manufacturing enterprises*. Kluwer Academic Publishers, Boston, MA, USA.

Viswanadham, N. and R.S. Gaonkar (2003). Partner selection and synchronized planning in dynamic manufacturing networks. *IEEE Trans. on Robotics and Automation* **19**(1), 117–130.

Viswanadham, N. and S. Raghavan (2000). Performance analysis and design of supply chains: a petri net approach. *Journal of the Operational Research Society* **51**, 1158–1169.

Wolter, K. (1997). Second order fluid stochastic Petri nets: an extention of gspns for approximate and continuous modelling. Proc. World Congr. on Systems Simulation. Singapore.

Wolter, K. and G. Hommel (1997). Hybrid modelling with second order fluid stochastic Petri nets. Proc. Workshop on Parallel and Distributed Real-Time Systems. Geneva, Switzerland.

Wolter, K. and G. Hommel (1998). Probabilistic and deterministic fluid flow in second order fspn's. Proc. 3rd Int. Conf. on Automation of Mixed Processes. Reims, France.

Zerhouni, C. Amer-Yahia; N. (2001). State equation and stability for a class of continuous Petri nets. application to the control of a production systems.. *Studies in Informatics and Control* **10**(4), 301–317.

Zerhouni, N. and H. Alla (1990). Dynamic analysis of manufacturing systems using continuous Petri nets. Proc. 1990 IEEE Int. Conf. on Robotics and Automation. Cincinnati, OH, USA.

Zerhouni, N. and H. Alla (1992). Manufacturing lines analysis using continuous Petri nets. *Autom. Prod. Inform. Ind.* **26**(3), 253–276.

Zhou, M.C. and K. Venkatesh (1998). *Modeling, Simulation and Control of Flexible Manufacturing Systems*. World Scientific, Singapore.