Modeling and Simulation of Manufacturing Systems with First–Order Hybrid Petri Nets

Fabio Balduzzi[*], Alessandro Giua[†], Carla Seatzu[‡]

Abstract

First–Order Hybrid Petri Nets are a model that consists of continuous places holding fluid, discrete places containing a non–negative integer number of tokens, and transitions, either discrete or continuous. In the first part of the paper, we provide a framework to describe the overall hybrid net behaviour that combines both time–driven and event–driven dynamics. The resulting model is a linear discrete–time, time–varying state variable model, that can be directly used by an efficient simulation tool. In the second part of the paper, we focus on manufacturing systems. Manufacturing systems are discrete event dynamic systems whose number of reachable states is typically very large, hence approximating fluid models have often been used in this context. We describe the net models of the elementary components of an FMS (machines and buffers) and we show in a final example how these modules can be put together in a bottom–up fashion.

Published as:

F. Balduzzi, A. Giua, C. Seatzu, "Modelling and Simulation of Manufacturing Systems Using First-Order Hybrid Petri Nets," *Int. J. of Production Research*, Vol. 39, No. 2, pp. 255-282, Jan 2001.

[*]Dip. Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino. Italy, balduzzi@polito.it

[†]Dip. Ingegneria Elettrica ed Elettronica, Università di Cagliari, Piazza d'Armi, 09123 Cagliari, Italy. giua@diee.unica.it

[‡]Dip. Ingegneria Elettrica ed Elettronica, Università di Cagliari, Piazza d'Armi, 09123 Cagliari, Italy. seatzu@diee.unica.it

1 Introduction

In this paper we use *First–Order Hybrid Petri Nets* (FOHPN), a class of nets that combines time–driven and event–driven dynamics, to describe manufacturing systems. This hybrid model, that was originally presented in (Balduzzi et al. 1999, 2000), answers a need — deeply felt in the field of manufacturing systems — for a formal tool that integrates the different phases of design, analysis and control of dynamical systems.

1.1 Motivation

The analysis and optimization of discrete event models require large amount of computational efforts. Thus, manufacturing systems are difficult to analyze and control and, with very few exceptions, problems of realistic scale quickly become analytically and computationally intractable.

To cope with this problem, *fluid models* which are continuous–dynamics approximations of discrete systems, have been successfully developed and applied by many authors (Caramanis 1987, Sharifnia 1994, Suri and Fu 1994, Balduzzi and Menga 1998) to the manufacturing domain. There are several advantages in using fluid approximations for analysis and control of complex manufacturing systems. First, there is the possibility of considerable increase in computational efficiency, because the simulation of fluid models can often be done much more efficiently. Second, fluid approximations provide an aggregate formulation to deal with complex systems, thus reducing the dimension of the state space. Their simple structures allow explicit computation and performance optimization. Third, the design parameters in fluid models are continuous (e.g. buffer sizes), hence it is possible to use gradient information to speed up optimization and to perform sensitivity analysis. Furthermore, it has also been shown that fluid approximations do not introduce significant errors when making performance analysis of manufacturing systems via simulations: the main performance measures are still surprisingly accurate (Suri and Fu 1994, Balduzzi and Menga 1998), given the fundamental differences in the dynamics of these models.

It should be noted that in general different fluid approximations are necessary to describe the same manufacturing system, depending on its discrete state: 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 dynamics is associated to each discrete state.

Petri nets (PN) (Murata 1989) have originally been introduced to describe and analyze discrete event systems. Recently, much effort has been devoted to apply these models to hybrid systems (Trivedi and Kulkarni 1993, David 1997, Alla and David 1998, Demongodin and Koussoulas 1998). The hybrid net model we consider in this paper, First–Order Hybrid Petri Nets, is general enough to model classes of hybrid systems of practical interest whose first–order continuous behaviour can be studied by linear algebraic tools and can be used to solve problems of myopic optimization and sensitivity analysis (Balduzzi et al. 2000). The aim of the paper is to show that FOHPN are extremely suited to model manufacturing systems where the arrival/departure of parts is described by fluid approximations.

1.2 The proposed model

FOHPN are a model that consists 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 FOHPN we distinguish two behavioral levels: time–driven and event–driven.

The continuous time–driven evolution of the net is described by a first–order fluid model, i.e. a model in which the *instantaneous firing speeds* (IFS) of the continuous transitions are piece–wise constant control variables which represent the machine production rates, the arrival of parts, and so on. Each IFS is a real vector $\boldsymbol{v}$ that may be chosen by the system operator within a given set $\mathcal{S}$ and represents a particular mode of operation of the system. Among all possible modes of operation, the system operator may choose the best according to a given objective (Balduzzi et al. 2000).

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 interval of time between the occurrence of two consecutive macro–events is called *macro–period*.

In (Balduzzi et al. 2000) the authors have considered two types of macro–events: (a) the firing of a discrete transition (that may represent the failure/repair of a machine); (b) the emptying of a continuous place (that may represent the emptying/filling of a buffer). However, the timing structure associated to the macro–event occurrence has not been explicitly examined in (Balduzzi et al. 2000). In this paper, we use timers to describe the timing structure associated to the transition firings. This implies that the set of macro–events has to be augmented to take into account those events that modify the timer values.

The overall hybrid net behaviour, that combines both time–driven and event–driven dynamics, can be described by a linear discrete–time, time-varying state variable model of the form $\boldsymbol{x}(k+1) = \boldsymbol{A}(k)\boldsymbol{x}(k) + \boldsymbol{B}(k)\boldsymbol{u}(k)$. The state vector $\boldsymbol{x}(k)$ at the end of the k-th macro–period is given by the marking of all places (continuous and discrete) and by the value of all timers associated to timed transitions. The input vector $\boldsymbol{u}(k)$ is given by the length of the k-th macro–period and by the characteristic vector that specifies which transition (if any) fires at the end of the k-th macro–period. We shall provide a simple algorithm to determine the state vector evolution at the end of each macro–period.

We see three main advantages in the proposed formulation. First, the linear state variable model can be directly implemented to construct an efficient and general simulation tool that may be used in the manufacturing domain. Second, this algebraic formalism allows one to describe manufacturing systems with a well–understood linear (albeit time–varying) state variable model to which classical control theory may be applied. Third, although we do not discuss this point in this paper, we recall that the FOHPN model is amenable to sensitivity analysis, i.e. it can be used to obtain information about the degrees of freedom that can be exploited when making performance optimization or optimal design of the system parameters configuration (Balduzzi et al. 2000).

The paper is structured as follows. In the first part we recall the definition of FOHPN and we present some useful tools for simulating its evolution. In particular, we provide a simple algorithm to determine the overall net behaviour that combines both time–driven and event–driven dynamics. In the second part, we focus on manufacturing systems. We describe the FOHPN models of the elementary components of an FMS (machines and buffers) and we show in a final example how these modules can be put together in a bottom–up fashion. Finally, we apply the results we have developed to a manufacturing system characterized by unreliable machines, buffers of finite capacity, different classes of products, and general routing policies.

2 Relevant literature

Fluid models have been successfully applied by many authors to the manufacturing domain. Sharifnia (1994) investigated the stability and performance of distributed control policy for continuous–flow production systems. Caramanis (1987) combined simulations of continuous tandem lines and non–linear programming techniques to solve optimization problems. Suri and Fu (1994) described the dynamics of continuous tandem lines in a form that enables GSMP representation. Other results related to the simulation of tandem production lines can be found in Phillis and Kouikoglou (1994), where a simulation algorithm much more efficient than conventional discrete event simulators was proposed. In a recent work Balduzzi and Menga (1998) developed a discrete–time, time–varying linear stochastic state variable model for the fluid approximation of flexible manufacturing systems. Then, by using perturbation analysis techniques they obtained average values and variances of both performance measures and their gradients with respect to the system parameters to perform optimal design of the system configuration.

The hybrid Petri net model we propose follows the formalism described by David and Alla (1998a, b). In effect our model can be seen as an extension of timed hybrid nets with constant maximal speeds with the following additional features: token reservation is not used; stochastic transitions are also included in our model; minimal firing speeds are considered as well. The novel contribution of our work is that of showing how the first–order behaviour of such a net can be efficiently described with a linear algebraic formalism, by exploiting the results of the fluid approximation theory. We originally introduced these ideas in (Balduzzi et al. 2000), where the basic concepts of FOHPN were presented, and in (Balduzzi et al. 1999), where the simulation algorithm to represent the overall net behaviour was given.

Linear algebraic techniques have also been used by Amrah et al. (1998) when modeling manufacturing systems with continuous Petri nets. These authors deal with open and closed transfer lines modelled by *controlled variable speed continuous Petri nets*, a type of continuous Petri nets (Alla and David 1998) with controllable maximal firing speeds. Then, by using a constrained optimization approach, they obtained optimal values for the machine production rates that bring the average levels of buffers to a desired value. However, it must be observed that transfer lines are not a general model, in the sense they do not require scheduling and routing strategies. Our framework, instead, may deal with manufacturing systems in the more general configurations and settings.

3 First–Order Hybrid Petri Nets

In this section we recall the Petri net formalism used in this paper following (Balduzzi et al. 2000). For a more comprehensive introduction to place/transition Petri nets see (Murata 1989). The common notation and semantics for timed nets can be found in (Ajmone et al. 1995).

3.1 Net structure

A First–Order Hybrid Petri Net (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 $\mathbb{R}_0^+ = \mathbb{R}^+ \cup \{0\}$):

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

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 $\mathcal{D} : T_t \rightarrow \mathbb{R}^+$ 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 $\mathcal{C} : T_c \rightarrow \mathbb{R}_0^+ \times \mathbb{R}_\infty^+$ 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 $\boldsymbol{C}_{dc} = \mathbf{0}_{n_d \times q_c}$.

Example 1. Consider the net in figure 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 (resp., taken from) the buffer.

The incidence matrix of this net is

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

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

[Insert figure 1 about here]

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 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. *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 4. In the net in figure 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.

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 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 (Ajmone 1995), 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/disabling a timed transition; (b) a continuous place reaches a fluid level that enables/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 $\hat{\nu}$'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 $\hat{\nu} = 0$.
Definition 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.

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 $\overline{V}_j \leq V_j$ 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/disabling a continuous transition; (b) a continuous place becomes empty, thus changing the enabling state of a continuous transition from strong to weak.

Definition 7 (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 8. In the net in figure 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 $\boldsymbol{\nu}(\tau) = [\tau - \tau_0,\ 0,\ \tau - \tau_0,\ 0]^T$. ■

4 Computation of a firing speed vector

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, and among all possible modes of operation, the system operator may choose the best according to a given objective.

4.1 Admissible IFS vectors

They form a convex set described by linear equations.

Definition 9 (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 $\boldsymbol{m}$, and $P_{\mathcal{E}} = \{p \in P_c \mid m_p = 0\}$ 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:*

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

Thus 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 10. Let $\langle N, \boldsymbol{m}(\tau_0) \rangle$ be the continuous net in figure 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:

$$\left\{ \begin{array}{ll} V_1 - v_1 & \geq 0 \\ V_2 - v_2 & \geq 0 \\ v_1, v_2 & \geq 0 \end{array} \right. \tag{7}$$

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. ∎

4.2 Optimal IFS vector

Once the set of all admissible IFS vectors has been defined, we need a procedure to select one among them. One possible way of computing an optimal IFS vector consists in introducing an objective function that may be representative of a global performance index and solving the corresponding optimization problem with the constraint set given by (6). We consider some examples.

- **Maximize flows**. In an FOHPN we may consider as optimal the solution $\boldsymbol{v}^*$ of (6) that maximizes the performance index $J = \boldsymbol{1}^T \cdot \boldsymbol{v}$ 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 $J = \boldsymbol{a}^T \cdot \boldsymbol{v}$ where

$$a_j = \left\{ \begin{array}{ll} 1 & \text{if } t_j \text{ is an exogenous transition,} \\ 0 & \text{if } t_j \text{ is an endogenous transition.} \end{array} \right.$$

  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 $J = \boldsymbol{a}^T \cdot \boldsymbol{v}$ where

$$a_j = \left\{ \begin{array}{ll} \boldsymbol{C}(p, t_j) & \text{if } t_j \in p^{(c)} \cup {}^{(c)}p, \\ 0 & \text{otherwise.} \end{array} \right.$$

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

4.3 Discussion

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

The definition of continuous transitions enabling proposed in (Alla and David 1998b) 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 figure 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 in (2001) a new concept, that of ϵ-marking: if an arbitrary small marking is initially assigned to any of the two places of the cycle in figure 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, that will be used to model *zero–capacity* buffers (see section 6.1). However, one may also adopt for FOHPN the enabling definition used in (David and Alla 2001).

Another difference with (Alla and David 1998b) 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 figure 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.

[Insert figure 2 about here]

5 Macro–behaviour: a linear model

In this section we show how it is possible to combine the continuous and discrete–event dynamics described in the previous section to obtain a linear time–varying, discrete–time state variable model.

5.1 Macro–events, macro–periods and state vector

If we consider both continuous and discrete–event dynamics, the system evolution is driven by four types of macro–events.

- π_i: a continuous place p_i becomes empty. This may change the enabling state of a set of continuous transitions from strong to weak, thus modifying the set $\mathcal{S}$.

- γ_j: a discrete transition t_j fires. This changes the discrete marking and may enable/disable a set of continuous transitions, thus modifying the set $\mathcal{S}$, or enable/disable a set of discrete transitions, thus modifying the vector of timers.

- ε_i: a continuous place p_i whose marking is increasing, reaches a flow level that enables a set of discrete transitions. This will enable the corresponding timers.

- $\bar{\varepsilon}_i$: a continuous place p_i whose marking is decreasing, reaches a flow level that disables a set of discrete transitions. This will disable and reset the corresponding timers.

In (Balduzzi et al. 2000) only the first two types of macro-events have been taken into account, since they are the only ones to produce a variation on $\mathcal{S}$.

The possible effects of each macro-event on the marking $\boldsymbol{m}$, the timer vector $\boldsymbol{\nu}$, the constraint set $\mathcal{S}$ are summarized in the table below.

	Jump in $\boldsymbol{m}$	Jump in $\boldsymbol{\nu}$	Change in $\mathcal{S}$	Timer	
				enab	disab
π_i			×		
γ_j	×	×	×	×	×
ε_i				×	
$\bar{\varepsilon}_i$		×			×

Let τ_k, for $k = 0, 1, 2, \ldots$, be the occurrence time of the k–th macro–event. The interval $[\tau_k, \tau_{k+1}]$ is called a *macro–period* and its length is denoted $\Delta(k+1) = (\tau_{k+1} - \tau_k)$. Note that a macro–period may have a null length whenever an immediate transition fires. As an example, suppose that a continuous place p_i reaches a fluid level that enables an immediate transition t_j. Then, the sequence of events will be 'ε_i at time τ_k' and 'γ_j at time τ_{k+1}', with $\tau_k = \tau_{k+1}$. This is similar to the notion of *vanishing state* in (Aimone 1995).

Our aim is that of obtaining a discrete–time state variable model of the system where each sampling instant k corresponds to the occurrence of the k–th macro–event, i.e. to the time instant τ_k. The overall state of the system is given by the marking of all places and by the values of all timers. Because of the choice of the single–server semantics only one timer is associated to each timed transition. Thus, we can define the state vector of the system as

$$\boldsymbol{x}(k) = \begin{bmatrix} \boldsymbol{m}^c(\tau_k) \\ \boldsymbol{m}^d(\tau_k) \\ \boldsymbol{\nu}(\tau_k) \end{bmatrix} \begin{matrix} \} \, n_c \\ \} \, n_d \\ \} \, q_t \end{matrix} \tag{8}$$

i.e. $\boldsymbol{x}(k) \in \mathbb{R}^s$, where $s = n_c + n_d + q_t$.

5.2 Discrete–time dynamics

We now derive a discrete–time state equation of the form:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}(k)\boldsymbol{x}(k) + \boldsymbol{B}(k)\boldsymbol{u}(k), \tag{9}$$

where $\boldsymbol{u}(k) \in \mathbb{R}_0^+ \times \mathbb{N}^{q_d}$ is the input vector, $\boldsymbol{A}(k)$ and $\boldsymbol{B}(k)$ are matrices of appropriate dimensions.

To show this, we first observe that the behaviour of an FOHPN can be described within a macro–period $[\tau_k, \tau_{k+1}]$ by the following equations:

$$\left\{ \begin{array}{lcl} \boldsymbol{m}^c(k+1) & = & \boldsymbol{m}^c(k) + \boldsymbol{C}_{cc}\boldsymbol{v}(k)\Delta(k+1) + \boldsymbol{C}_{cd}\boldsymbol{\sigma}(k+1) \\ \boldsymbol{m}^d(k+1) & = & \boldsymbol{m}^d(k) + \boldsymbol{C}_{dd}\boldsymbol{\sigma}(k+1) \\ \boldsymbol{\nu}(k+1) & = & \boldsymbol{D}(k)\boldsymbol{\nu}(k) + \boldsymbol{f}(k)\Delta(k+1). \end{array} \right. \tag{10}$$

where we have written all vectors as functions of k instead of τ_k. Here $\boldsymbol{\sigma}(k+1)$ is the firing count vector that specifies which discrete transition fires at time τ_{k+1} and $\Delta(k+1)$ is the length of the macro–period.

The first two equations follow from the combination of the net dynamic equations (3), (5). The third equation follows from (1) and (2). Here matrix $\boldsymbol{D}(k) \in \{0,1\}^{q_t \times q_t}$ and vector $\boldsymbol{f}(k) \in \{0,1\}^{q_t}$ depend on the macro–event occurring at the sampling instant $k+1$. In particular, the following definitions, whose validity can be easily verified, hold:

- *Macro-events* π_i, ε_i. There will be no jump in $\boldsymbol{\nu}$ and each timer associated to an enabled transition increases with unitary rate. Therefore for all $t_h \in T_t$

$$f_{\varrho(h)}(k) = \left\{ \begin{array}{ll} 1 & \text{if } t_h \text{ is enabled at } \boldsymbol{m}(k) \\ 0 & \text{otherwise} \end{array} \right. \quad ; \qquad \boldsymbol{D}(k) = \text{diag}\{\boldsymbol{f}(k)\}, \tag{11}$$

  i.e. $\boldsymbol{D}(k)$ is a diagonal matrix with entries $D_{h,h} = f_h$.

- *Macro-event* γ_j. Let $\overline{T}_j \subset T_t$ be the subset of timed transitions disabled by the firing of t_j. For all $t_h \in \overline{T}_j$, $\nu_h(k+1)$ will be reset to 0 regardless of the value of $\nu_h(k)$. Furthermore the timer of t_j will be also reset to 0 after its firing. Therefore for all $t_h \in T_t$

$$f_{\varrho(h)}(k) = \left\{ \begin{array}{ll} 1 & \text{if } (j \neq h) \text{ and } (t_h \notin \overline{T}_j) \\ & \text{and } t_h \text{ is enabled at } \boldsymbol{m}(k) \\ 0 & \text{otherwise} \end{array} \right. \quad ; \qquad \boldsymbol{D}(k) = \text{diag}\{\boldsymbol{f}(k)\}. \tag{12}$$

- *Macro-event* $\bar{\varepsilon}_i$. Let $\overline{T}_i \subset T_t$ be the subset of timed transitions disabled by the decreasing marking of the continuous place p_i. The timers of all these transitions will be reset to 0. Therefore for all $t_h \in T_t$

$$f_{\varrho(h)}(k) = \left\{ \begin{array}{ll} 1 & \text{if } (t_h \notin \overline{T}_i) \text{ and} \\ & t_h \text{ is enabled at } \boldsymbol{m}(k) \\ 0 & \text{otherwise} \end{array} \right. \quad ; \qquad \boldsymbol{D}(k) = \text{diag}\{\boldsymbol{f}(k)\}. \tag{13}$$

Finally, we observe that equation (10) is in the form of equation (9) if we consider (8) and let

$$\boldsymbol{A}(k) = \begin{bmatrix} \boldsymbol{I}_{n_c \times n_c} & \boldsymbol{0}_{n_c \times n_d} & \boldsymbol{0}_{n_c \times q_t} \\ \boldsymbol{0}_{n_d \times n_c} & \boldsymbol{I}_{n_d \times n_d} & \boldsymbol{0}_{n_d \times q_t} \\ \boldsymbol{0}_{q_t \times n_c} & \boldsymbol{0}_{q_t \times n_d} & \boldsymbol{D}(k) \end{bmatrix}, \quad \boldsymbol{B}(k) = \begin{bmatrix} \boldsymbol{C}_{cc}\boldsymbol{v}(k) & \boldsymbol{C}_{cd} \\ \boldsymbol{0}_{n_d \times 1} & \boldsymbol{C}_{dd} \\ \boldsymbol{f}(k) & \boldsymbol{0}_{q_t \times n_d} \end{bmatrix}, \quad \boldsymbol{u}(k) = \begin{bmatrix} \Delta(k+1) \\ \boldsymbol{\sigma}(k+1) \end{bmatrix}. \tag{14}$$

The input vector $\boldsymbol{u}$ specifies: (a) the length $\Delta(k+1)$ of the current macro–period; (b) which transition (if any) will fire at the end of the current macro–period. Note that $\Delta(k+1)$ and $\boldsymbol{\sigma}(k+1)$ depend on the state vector $\boldsymbol{x}(k)$ and on the macro–event occurring at the end of the current macro–period. We can explicitly write their value as follows.

- *Macro-event* π_i. The length of the macro–period is the time it takes to empty the continuous place p_i, i.e. the ratio between its actual marking and its variation with respect to time (changed of sign). The firing count vector is equal to the null vector since no discrete transition fires. Therefore,

$$\Delta(k+1) = \frac{-m_i(k)}{\dot{m}_i(k)} = \frac{-\boldsymbol{e}_{i,s}^T \boldsymbol{x}(k)}{\boldsymbol{e}_{i,n_c}^T \boldsymbol{C}_{cc} \boldsymbol{v}(k)}, \qquad \boldsymbol{\sigma}(k+1) = \boldsymbol{0}_{q_d \times 1}. \tag{15}$$

- *Macro-event* γ_j. If t_j is a timed transition, the length of the macro–period is the residual lifetime of the transition timer, i.e.

$$\Delta(k+1) = \hat{\nu}_j - \nu_j(k) = \hat{\nu}_j - \boldsymbol{e}_{n+\varrho(j),s}^T \boldsymbol{x}(k), \tag{16}$$

else if t_j is an immediate transition, then $\Delta(k+1) = 0$. Finally,

$$\boldsymbol{\sigma}(k+1) = \boldsymbol{e}_{\varrho(j),q_d}. \tag{17}$$

- *Macro-events* ε_i, $\bar{\varepsilon}_i$. The length of the macro–period is the time it takes the marking m_i to reach the value $\boldsymbol{C}(p_i, t_j)$ thus enabling (disabling) some discrete transition t_j. As in the first case, the firing count vector is equal to the null vector, since no discrete transition fires. Therefore,

$$\Delta(k+1) = \frac{\boldsymbol{C}(p_i, t_j) - m_i(k)}{\dot{m}_i(k)} = \frac{\boldsymbol{C}(p_i, t_j) - \boldsymbol{e}_{i,s}^T \boldsymbol{x}(k)}{\boldsymbol{e}_{i,n_c}^T \boldsymbol{C}_{cc} \boldsymbol{v}(k)}, \qquad \boldsymbol{\sigma}(k+1) = \boldsymbol{0}_{q_d \times 1}. \tag{18}$$

5.3 A simulation algorithm

In this subsection we provide a simulation algorithm to determine the state vector at the beginning of each macro–period, given the initial state $\boldsymbol{x}(0)$. We assume that a criterion J to select the optimal IFS vector among all admissible ones is also given.

Algorithm 11. *Simulation algorithm.*

1. Let $k := 0$, $\boldsymbol{\nu}(\tau_0) = \boldsymbol{0}_{q_t \times 1}$, $\boldsymbol{m}(0) := \begin{bmatrix} \boldsymbol{m}^c(\tau_0) \\ \boldsymbol{m}^d(\tau_0) \end{bmatrix}$, and $\boldsymbol{x}(0) := \begin{bmatrix} \boldsymbol{m}^c(\tau_0) \\ \boldsymbol{m}^d(\tau_0) \\ \boldsymbol{\nu}(\tau_0) \end{bmatrix}$. Compute the value of each timer set point $\hat{\nu}$.

2. Select an IFS vector $\boldsymbol{v}(k) \in \mathcal{S}(N, \boldsymbol{m}(k))$ so as to optimize the chosen criterion J.

3. Determine the next macro–event to occur $\bar{\beta}$ according to the following steps.

(a) Let $\Psi_k := \emptyset$. (This set in step 3.(h) will contain all pairs (β, Δ_β), where β is an event that may potentially occur and Δ_β is its residual lifetime.)

(b) For each immediate transition t_j enabled at $\boldsymbol{x}(k)$, add to Ψ_k the pair $(\gamma_j, 0)$.

(c) If $\Psi_k \neq \emptyset$, then goto step 3.(h).

(d) For each timed transition t_j enabled at $\boldsymbol{x}(k)$, add to Ψ_k the pair $(\gamma_j, \hat{\nu}_j - \boldsymbol{e}_{\varrho(j)+n,s}^T \boldsymbol{x}(k))$.

(e) For each non-empty continuous place p_i, if $\dot{m}_i(k) = \boldsymbol{e}_{i,n_c}^T \boldsymbol{C}_{cc} \boldsymbol{v}(k) < 0$ then add to Ψ_k the pair $(\pi_i, \dfrac{-\boldsymbol{e}_{i,s} \boldsymbol{x}(k)}{\boldsymbol{e}_{i,n_c}^T \boldsymbol{C}_{cc} \boldsymbol{v}(k)})$.

(f) For each discrete transition t_j that is not enabled at $\boldsymbol{x}(k)$, let $P_j := \{p_\ell \in {}^{(c)}t_j \mid m_\ell(k) < \boldsymbol{C}(p_\ell, t_j)\}$ be the set of continuous places that have not enough fluid content to enable t_j. This transition may become enabled at the end of the current macro–period if the following two conditions are both verified.

- $\boldsymbol{m}^d(k) \geq \boldsymbol{C}_{dd}(\cdot, t_j)$, i.e. t_j is enabled in the discrete sub–net;
- $\forall p_\ell \in P_j$, $\dot{m}_\ell(k) = \boldsymbol{e}^T_{\ell,n_c}\boldsymbol{C}_{cc}\boldsymbol{v}(k) > 0$, i.e. the marking of all places in P_j is increasing.

The time it takes for t_j to become enabled is

$$\Delta := \max_{p_\ell \in P_j} \frac{\boldsymbol{C}(p_\ell, t_j) - \boldsymbol{e}_{\ell,s}\boldsymbol{x}(k)}{\boldsymbol{e}_{\ell,n_c}\boldsymbol{C}_{cc}\boldsymbol{v}(k)}$$

and we denote p_i the place for which this value is maximum. However, t_j will not be enabled if any place $p_{\bar{\ell}}$ in the set $\bar{P}_j := \{p_\ell \in {}^{(c)}t_j \mid m_\ell(k) \geq \boldsymbol{C}(p_\ell, t_j), \dot{m}_\ell(k) < 0\}$, will go below the fluid level $\boldsymbol{C}(p_{\bar{\ell}}, t_j)$ in the meantime. Thus we let

$$\bar{\Delta} := \min_{p_\ell \in \bar{P}_j} \frac{\boldsymbol{C}(p_\ell, t_j) - \boldsymbol{e}_{\ell,s}\boldsymbol{x}(k)}{\boldsymbol{e}_{\ell,n_c}\boldsymbol{C}_{cc}\boldsymbol{v}(k)}$$

with $\bar{\Delta} := \infty$ if $\bar{P}_j = \emptyset$. If $\bar{\Delta} > \Delta$, add (ε_i, Δ) to Ψ_k.

(g) For each enabled discrete transition t_j, let $\bar{P}_j := \{p_\ell \in {}^{(c)}t_j \mid \dot{m}_\ell(k) < 0\}$ be the set of continuous places whose marking is decreasing. If $\bar{P}_j \neq \emptyset$, transition t_j may become disabled. In this case let

$$\Delta := \min_{p_\ell \in \bar{P}_j} \frac{\boldsymbol{C}(p_\ell, t_j) - \boldsymbol{e}_{\ell,s}\boldsymbol{x}(k)}{\boldsymbol{e}_{\ell,n_c}\boldsymbol{C}_{cc}\boldsymbol{v}(k)},$$

and let p_i be the place corresponding to this minimum. The macro–event $\bar{\varepsilon}_i$ will occur not at time Δ, but an instant later when m_i will go below the value $\boldsymbol{C}(p_i, t_j)$. Thus we add $(\bar{\varepsilon}_i, \Delta^+)$ to Ψ_k.

(h) Choose from Ψ_k the pair $(\bar{\beta}, \Delta_{\bar{\beta}})$ where $\Delta_{\bar{\beta}}$ is the minimum over all pairs. Event $\bar{\beta}$ is the next to occur. Note that if two pairs (β_1, Δ) and (β_2, Δ^+) are in Ψ_k, then $(\Delta < \Delta^+)$ and β_1 should be chosen.

4. Let $\boldsymbol{x}(k+1) := \boldsymbol{A}(k)\boldsymbol{x}(k) + \boldsymbol{B}(k)\boldsymbol{u}(k)$ where matrices $\boldsymbol{A}(k)$, $\boldsymbol{B}(k)$ and vector $\boldsymbol{u}(k)$ are defined in accordance with the results in subsection 5.2 and depend on the type of macro-event $\bar{\beta}$.

   Let $k := k + 1$, and $\boldsymbol{m}(k) := [\boldsymbol{I}_{n\times n}\ \boldsymbol{0}_{n\times q_t}]\ \boldsymbol{x}(k)$.

5. Update the value of each timer set point $\hat{\nu}$. The set point of all timed transitions disabled by macro-event $\bar{\beta}$ should be reset to 0. The set point of all transitions enabled after the occurrence of macro-event $\bar{\beta}$ should be reinitialized. Goto step 2. ■

Now, let us apply the above algorithm to an example.

Example 12. Consider the net in figure 1.a, already discussed in the previous examples, and let the initial time be $\tau_0 = 0$. Let the initial marking be that represented in the figure. We assume that at each macro–period the performance index to maximize is the overall machines utilization, i.e. $J = v_1 + v_2$.

k=0. The initial state vector is $\boldsymbol{x}(0) = [m_1,\ 1,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0]^T$. The enabled stochastic transitions are t_3 and t_5, so we extract the timer set points $\hat{\nu}_3$ and $\hat{\nu}_5$ from the corresponding distribution. Assume $\hat{\nu}_5 < \hat{\nu}_3$.

Transitions t_1 and t_2 are strongly enabled and may fire at their maximum speeds, thus to maximize J we choose $\boldsymbol{v}(0) = [V_1,\ V_2]^T$.

We compute $\Psi_0 = \{(\pi_1, \Delta_{\pi_1}),\ (\gamma_3, \Delta_{\gamma_3}),\ (\gamma_5, \Delta_{\gamma_5})\}$, where $\Delta_{\pi_1} = m_1/(V_2\ b - V_1\ a)$, $\Delta_{\gamma_3} = \hat{\nu}_3$ and $\Delta_{\gamma_5} = \hat{\nu}_5$.

Assume $\Delta_{\pi_1} < \hat{\nu}_5$. Then π_1 is the next event to occur, i.e. the first macro–event is due to the emptying of continuous place p_1. The length of the first macro–period is $\Delta(1) = \Delta_{\pi_1}$ and the occurrence time of the first macro–event is $\tau_1 = \Delta_{\pi_1}$.

Matrices $\boldsymbol{A}(0)$, $\boldsymbol{B}(0)$, and the input vector $\boldsymbol{u}(0)$ have the following values:

$$\boldsymbol{A}(0) = \begin{bmatrix} 1 & \mathbf{0}_{1\times 4} & \mathbf{0}_{1\times 4} \\ \mathbf{0}_{4\times 1} & \boldsymbol{I}_{4\times 4} & \mathbf{0}_{4\times 4} \\ \mathbf{0}_{4\times 1} & \mathbf{0}_{4\times 4} & \mathrm{diag}\{[1\,0\,1\,0]\} \end{bmatrix}, \quad \boldsymbol{B}(0) = \begin{bmatrix} V_1 - V_2 & \mathbf{0}_{1\times 4} \\ \mathbf{0}_{4\times 1} & \boldsymbol{C}_{dd} \\ [1\,0\,1\,0]^T & \mathbf{0}_{4\times 4} \end{bmatrix}, \quad \boldsymbol{u}(0) = \begin{bmatrix} \tau_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

k=1. The state vector at the beginning of the second macro–period is $\boldsymbol{x}(1) = [0,\ 1,\ 0,\ 1,\ 0,\ \tau_1,\ 0,\ \tau_1,\ 0]^T$. The timer set points keeps the previous values, because no timed transition changes its enabling state.

During this macro–period, t_1 remains strongly enabled, while t_2 is weakly enabled and may fire at most at speed $\overline{V}_2 = v_1 \cdot (a/b) < V_2$. Thus, the optimal IFS vector is $\boldsymbol{v}(1) = [V_1,\ V_1 \cdot (a/b)]^T$.

We compute $\Psi_1 = \{(\gamma_3, \Delta_{\gamma_3}),\ (\gamma_5, \Delta_{\gamma_5})\}$, where $\Delta_{\gamma_3} = \hat{\nu}_3 - \tau_1$ and $\Delta_{\gamma_5} = \hat{\nu}_5 - \tau_1$.

Then γ_5 will be the next event to occur, i.e. the second macro–event is due to the firing of transition t_5. The length of the second macro–period is $\Delta(2) = \Delta_{\gamma_5} = \hat{\nu}_5 - \tau_1$ and the occurrence time of the second macro–event is $\tau_2 = \Delta_{\gamma_5} + \tau_1 = \hat{\nu}_5$.

We compute

$$\boldsymbol{A}(1) = \begin{bmatrix} 1 & \mathbf{0}_{1\times 4} & \mathbf{0}_{1\times 4} \\ \mathbf{0}_{4\times 1} & \boldsymbol{I}_{4\times 4} & \mathbf{0}_{4\times 4} \\ \mathbf{0}_{4\times 1} & \mathbf{0}_{4\times 4} & \mathrm{diag}\{[1\,0\,0\,1]\} \end{bmatrix}, \quad \boldsymbol{B}(1) = \begin{bmatrix} V_1(1 - a/b) & \mathbf{0}_{1\times 4} \\ \mathbf{0}_{4\times 1} & \boldsymbol{C}_{dd} \\ [1\,0\,0\,1]^T & \mathbf{0}_{4\times 4} \end{bmatrix}, \quad \boldsymbol{u}(1) = \begin{bmatrix} \tau_2 - \tau_1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

k=2. The state vector at the beginning of the third macro–period is $\boldsymbol{x}(2) = [0,\ 1,\ 0,\ 0,\ 1,\ \tau_2,\ 0,\ 0,\ 0]^T$. The timer set point of transition t_3 is kept, while the one of transition t_5 is reset to 0. Transition t_6 has become enabled, so we extract the timer set points $\hat{\nu}_6$ from the corresponding distribution. Assume $\hat{\nu}_3 - \tau_2 < \hat{\nu}_6$.

The optimal IFS vector is $\boldsymbol{v}(2) = [V_1,\ 0]^T$, since only continuous transition t_1 is enabled and may fire at its maximum firing speed.

We compute $\Psi_2 = \{(\gamma_3, \Delta_{\gamma_3}),\ (\gamma_6, \Delta_{\gamma_6})\}$, where $\Delta_{\gamma_3} = \hat{\nu}_3 - \tau_2$ and $\Delta_{\gamma_6} = \hat{\nu}_6$.

Then γ_3 will be the next event to occur, i.e. the third macro–event is due to the firing of transition t_3. The length of the third macro–period is $\Delta(3) = \Delta_{\gamma_3} = \hat{\nu}_3 - \tau_2$ and the occurrence time of the third macro–event is $\tau_3 = \Delta_{\gamma_3} + \tau_2 = \hat{\nu}_3$.

We compute

$$\boldsymbol{A}(2) = \begin{bmatrix} 1 & \mathbf{0}_{1\times4} & \mathbf{0}_{1\times4} \\ \mathbf{0}_{4\times1} & \boldsymbol{I}_{4\times4} & \mathbf{0}_{4\times4} \\ \mathbf{0}_{4\times1} & \mathbf{0}_{4\times4} & \text{diag}\{[0\,1\,0\,1]\} \end{bmatrix}, \quad \boldsymbol{B}(2) = \begin{bmatrix} V_1 & \mathbf{0}_{1\times4} \\ \mathbf{0}_{4\times1} & \boldsymbol{C}_{dd} \\ [0\,1\,0\,1]^T & \mathbf{0}_{4\times4} \end{bmatrix}, \quad \boldsymbol{u}(2) = \begin{bmatrix} \tau_3 - \tau_2 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

And so on. ■

6 Modeling manufacturing systems with FOHPN

In this section we show how FOHPN can be used to model manufacturing systems producing a variety of products and consisting of unreliable machines and buffers in a general network configuration.

We define the basic modules for buffers and machines and the interconnections between them. These modules can be composed in a bottom-up fashion, as shown in the example in the next section. Note that we consider here only continuous–flow production systems. However, using discrete transitions is also possible to model batch systems in which batches of products are processed at a time.

6.1 Buffer models

Let r be the number of different part classes in the system. A *multi–class* (MC) buffer B_i is modeled with r continuous places p^q_{Bi}, for $q = 1, \ldots, r$, whose marking represents the buffer content of parts of class q.

Let $I^q_{in,Bi}$ and $I^q_{out,Bi}$ be the set of indexes of machines that may, respectively, deposit or take from the buffer parts of class q. The arrival of parts of class q from machine M_j, where $j \in I^q_{in,Bi}$, is modeled by a continuous transition labeled $t^q_{Mj,Bi}$ inputting into p^q_{Bi}. The routing of parts of class q to machines M_j, where $j \in I^q_{out,Bi}$, is modeled by a continuous transition labeled $t^q_{Bi,Mj}$ outputting from p^q_{Bi}. These transitions represent the interfaces among machines and buffers: if no constraint is associated to these flows, the MFS of these transitions are taken to be ∞.

If the buffer has a finite capacity C_{Bi}, then a continuous place $\overline{p}_{Bi}$ will also be present in the buffer model. This new place will have arcs $Pre(\overline{p}_{Bi}, \cdot) = \sum_{q=1}^r Post(p^q_{Bi}, \cdot)$ and $Post(\overline{p}_{Bi}, \cdot) = \sum_{q=1}^r Pre(p^q_{Bi}, \cdot)$. The initial marking of place $\overline{p}_{Bi}$ is chosen as: $m_{\overline{p}_{Bi}}(\tau_0) = C_{Bi} - \sum_{q=1}^r m_{p^q_{Bi}}(\tau_0)$. Thus for any reachable marking $\boldsymbol{m}$ holds $m_{\overline{p}_{Bi}} + \sum_{q=1}^r m_{p^q_{Bi}} = C_{Bi}$.

The FOHPN model of a finite capacity MC buffer is shown in figure 3. The initial marking shown assumes that the buffer is initially empty. For each MC buffer the following set of equations will be included in $\mathcal{S}(N, \boldsymbol{m})$:

$$\begin{cases} (a) & \sum_{j \in I^q_{in,Bi}} v^q_{Mj,Bi} & \geq & \sum_{j \in I^q_{out,Bi}} v^q_{Bi,Mj} & \text{if} \quad m_{p^q_{Bi}} = 0 \\ (b) & \sum_{q=1}^r \sum_{j \in I^q_{out,Bi}} v^q_{Bi,Mj} & \geq & \sum_{q=1}^r \sum_{j \in I^q_{in,Bi}} v^q_{Mj,Bi} & \text{if} \quad m_{\overline{p}_{Bi}} = 0 \end{cases} \tag{19}$$

For each class q, if the buffer does not contain parts of this class, i.e. $m_{p^q_{Bi}} = 0$, a constraint of type (19.a) is used. If the buffer is full, i.e. $m_{\overline{p}_{Bi}} = 0$, then constraint (19.b) is also used. Depending on the current marking $\boldsymbol{m}$, the number of constraints associated to an MC buffer may vary from 0 — when the buffer is not full but contains parts of all classes — to r — when the buffer is full of parts of only one class.

A *single–class buffer* B_i reduces to a simpler model with only two continuous places, denoted p_{Bi} and $\overline{p}_{Bi}$.

[Insert figure 3 about here]

Zero–capacity buffer

Sometimes it may be necessary to impose synchronization constraints among continuous transitions. As an example, if we want to say that the overall flow of transitions t_1 and t_2 is equal to the overall flow of transitions t_3, t_4 and t_5 we require that

$$v_1 + v_2 \quad = \quad v_3 + v_4 + v_5.$$

This can be done, as in figure 4, introducing a *zero–capacity* buffer, represented by the empty continuous places p and $\overline{p}$. These two empty places enforces two constraints of the form given in Eq. (6.d) on the IFS of their input and output continuous transitions:

$$\begin{cases} v_1 + v_2 & \geq \quad v_3 + v_4 + v_5 \\ v_3 + v_4 + v_5 & \geq \quad v_1 + v_2 \end{cases}$$

Note that the buffer introduces an empty cycle. If the enabling definition of (David and Alla 2001) is used, an ϵ-marking should be assigned to this cycle (see section 4.3).

[Insert figure 4 about here]

6.2 Machine production models

The production of a *multi–class* (MC) machine M_i processing r classes of products is modeled with r single–class subnets. The subnet associated to each class q, for $q = 1, \ldots, r$, has a continuous transition t^q_{Mi} to represent the processing of parts of class q, and one input and one output zero–capacity buffer — represented by continuous places $p^q_{in,Mi}$, $\overline{p}^{\,q}_{in,Mi}$ and $p^q_{out,Mi}$, $\overline{p}^{\,q}_{out,Mi}$ — to impose that for parts of class q the total input flow is equal to the processed flow and to the total output flow.

Let $I^q_{in,Mi}$ and $I^q_{out,Mi}$ be the set of indexes of, respectively, input and output buffers of machine M_i for parts of class q. The interfaces among machine and buffers are represented by continuous transitions $t^q_{Bj,Mi}$ (for $j \in I^q_{in,Mi}$) and $t^q_{Mi,Bj}$ (for $j \in I^q_{out,Mi}$), as already discussed in the model of the buffers.

We assume that the production of any part class is not singularly bounded, i.e. the MFS of each t^q_{Mi} is ∞, but we assume that the machine has an overall production rate, modeled by the firing of transition t_{Mi}, that is bounded by V_{Mi}. Thus the continuous transition t_{Mi} is synchronized with all t^q_{Mi} by the zero–capacity buffer represented by continuous places p_{Mi} and $\overline{p}_{Mi}$: this ensures that $v_{Mi} = v^1_{Mi} + \cdots + v^r_{Mi}$.

As parts of different classes may require for their processing different service times, let us denote with θ_q the average service time of parts of class q, and let $\tilde{\theta} = \min_q\{\theta_q\}$. Thus we can assume $V_{Mi} = \tilde{\theta}^{-1}$ and defining $\gamma_q = \theta_q V_{Mi}$, for $q = 1, \ldots, r$, we obtain the FOHPN model shown in figure 5. For such a MC machine we have the following set of $(2r+2)$ equations:

$$\begin{cases} (a) & v_{Mi} & \leq & V_{Mi} & \\ (b) & v_{Mi} & = & \sum_{q=1}^{r} \gamma_q v_{Mi}^q & \\ (c) & v_{Mi}^q & = & \sum_{j \in I_{in,Mi}^q} v_{Bj,Mi}^q & (\forall\, q = 1, \ldots, r) \\ (d) & v_{Mi}^q & = & \sum_{j \in I_{out,Mi}^q} v_{Mi,Bj}^q & (\forall\, q = 1, \ldots, r) \end{cases} \tag{20}$$

Equation (20.b) derives from the zero–capacity buffer p_{Mi} and $\overline{p}_{Mi}$. The 2 r equations (20.c) and (20.d) derive from the input and output zero–capacity buffers of each class q subnet.

[Insert figure 5 about here]

Two simpler representation of the machine model are possible. In the case of a *single–class machine*, we have just one of the sub–nets represented within the continuous line boxes in figure 5. Clearly, the specification of the class is needless: the unique continuous transition of this sub–net will be simply labeled t_{Mi} and have an MFS V_{Mi}. The zero–capacity buffer p_{Mi} and $\overline{p}_{Mi}$ will be removed.

Another simplification is possible if part of class q are coming from just one input buffer B_h. In this case the zero–capacity buffer represented by places $p_{in,Mi}^q$ and $\overline{p}_{in,Mi}^q$ may be removed, directly connecting transition t_{Mi}^q to the buffer B_h. Similar reasonings can be applied when the machine is putting processed parts into a single buffer.

6.3 Machine failure models

We assume that machines may be unreliable and consider two different failure models. A *time–dependent failure* (TDF) model assumes that a machine fails after a given time has elapsed since the previous repair operation. An *operation–dependent failure* (ODF) model assumes that a machine fails after a given production volume has been processed since the previous repair operation. As suggested in (Buzacott and Hanifin 1978) the ODF model is more appropriate than the TDF model when dealing with manufacturing systems. However TDF models are suitable when programmed maintenance is adopted.

TDF models can be represented with an FOHPN as shown in figure 6. This model is similar to the one presented in (Alla and David 1998b). Continuous firing of transition t_{Mi} corresponds to a continuous production at rate $v_{Mi} \leq V_{Mi}$. The machine will keep on producing until it is *operational*, that is the place $p_{up,i}$ is marked. The firing of transition $t_{F,i}$ corresponds to the machine failure and this event occurs after a random delay exponentially distributed with parameter $\lambda_{F,i}$. When $t_{F,i}$ fires, the token in $p_{up,i}$ moves to place $p_{down,i}$, hence t_{Mi} is disabled and cannot fire. Analogously, the firing of transition $t_{R,i}$ corresponds to the machine repair. In figure 6 we have used stochastic transitions to represent the fail/repair events, but deterministic transitions may be used as well.

[Insert figure 6 about here]

ODF models can be represented with an FOHPN as shown in figure 7. The continuous place $p_{R,i}$ is initially marked with w_i that represents the production volume that will be processed by the machine M_i before failing. After machine M_i has processed the fluid quantity w_i, the continuous place $p_{F,i}$ will be marked by w_i. Then, the immediate transition $t_{down,i}$ is enabled and fires, emptying place $p_{F,i}$ and removing 1 token from place $p_{up,i}$, thus disabling transition t_{Mi}. At the same time 1 token is added to the discrete place $p_{down,i}$, thus enabling the stochastic transition $t_{R,i}$. Place $p_{down,i}$ represents the condition of the machine under repairing because when it is marked transition t_{Mi} is disabled, i.e. $v_{Mi} = 0$. The machine will be down until the repair event occurs, i.e. transition $t_{R,i}$ fires, bringing the net back to the initial state.

The ODF model exploits one of the hybrid features of FOHPN: the transformation of fluids into discrete tokens and vice versa through discrete transitions.

[Insert figure 7 about here]

6.4 A job–shop example

In this section we consider the FOHPN model of a production system consisting of a simple job–shop already presented in the literature (Sethi and Zhang 1994); we have only added an assembly operation at the end of the process. Its scheme is sketched in figure 8.

[Insert figure 8 about here]

This job–shop has 4 machines M_1, M_2, M_3, M_4, an assembly station M_5, and seven buffers, B_0, B_1, B_2, B_3, B_4, B_5, B_6. Machine M_i ($i = 1, 2, 3, 4, 5$) has a maximum machine production rate V_{Mi}. Buffers B_0 and B_6 are fictitious in the sense that B_0 is an infinite source containing all required raw materials and B_6 is a sink buffer with no constraints. On the contrary all other buffers have a finite capacity C_{Bi}.

There are two classes of products: class 1, whose flow is denoted in figure 8 by thin lines, and class 2, whose flow is denoted by thick lines. The flow of assembled product is denoted in figure 8 by a dotted line. Row parts of both classes are initially contained in B_0.

Initially, all buffers are empty (except the raw parts buffer B_0) and all machines are assumed to be operational. The FOHPN model of the production system under consideration is reported in figure 9.

[Insert figure 9 about here]

For the sake of simplicity, we assume that only M_1 and M_2 are unreliable machines characterized by a time–dependent failure model. Furthermore, machine M_2 has the same service time when processing parts of both classes. We also assume that all buffers have finite capacity except the two–class buffer B_0 and the final single–class buffer B_6.

We apply the simulation algorithm 11 to the job–shop example in figure 8, computing the state vector during four macro–periods. Numerical values of matrices $\boldsymbol{A}(k)$ and $\boldsymbol{B}(k)$ have not been reported here to keep the example short.

Let us consider the following numerical values: $V_{M1} = 10$, $V_{M2} = 15$, $V_{M3} = 30$, $V_{M4} = 10$, $V_{M5} = 10$; $\alpha_2 = 0.7$; $m^1_{B0} = 10^3$, $m^2_{B0} = 10^3$; $C_{B1} = 50$, $C_{B2} = 50$, $C_{B3} = 40$, $C_{B4} = 30$, $C_{B5} = 30$; $\lambda_{R,1} = 0.1$, $\lambda_{F,1} = 0.06$, $\lambda_{R,2} = 0.07$, $\lambda_{F,2} = 0.2$.

We denote the set of places as:

$$P = \{p_1, \cdots, p_{21}\} = \{p^1_{B0}, p^2_{B0}, p_{B1}, \overline{p}_{B1}, p_{B2}, \overline{p}_{B2}, p_{B3}, \overline{p}_{B3}, p_{B4}, \overline{p}_{B4}, p_{B5}, \overline{p}_{B5}, p_{B6}, p_{M4}, \overline{p}_{M4}, p_{in,M3}, \overline{p}_{in,M3}, p_{o,1}, p_{d,1}, p_{o,2}, p_{d,2}\};$$

the set of transitions as:

$$T = \{t_1, \cdots, t_{13}\} = \{t_{M1}, t_{M2}, t_{M3}, t^1_{M4}, t^2_{M4}, t_{M4}, t_{M5}, t_{B1,M3}, t_{B3,M3}, t_{R,1}, t_{F,1}, t_{R,2}, t_{F,2}\};$$

the state space vector at the beginning of the $(k+1)$–th macro–period as:

$$\boldsymbol{x}(k) = [\boldsymbol{m}^c(k)^T, \boldsymbol{m}^d(k)^T, \boldsymbol{\nu}(k)^T]^T,$$

where

$$\begin{aligned} \boldsymbol{m}^c(k) = [&m_{p^1_{B0}}(k), m_{p^2_{B0}}(k), m_{p_{B1}}(k), m_{\overline{p}_{B1}}(k), m_{p_{B2}}(k), m_{\overline{p}_{B2}}(k), \\ &m_{p_{B3}}(k), m_{\overline{p}_{B3}}(k), m_{p_{B4}}(k), m_{\overline{p}_{B4}}(k), m_{p_{B5}}(k), m_{\overline{p}_{B5}}(k), \\ &m_{p_{B6}}(k), m_{p_{M4}}(k), m_{\overline{p}_{M4}}(k), m_{p_{in,M3}}(k), m_{\overline{p}_{in,M3}}(k)]^T, \end{aligned}$$

$$\boldsymbol{m}^d(k) = [m_{p_{o,1}}, m_{p_{d,1}}, m_{p_{o,2}}, m_{p_{d,2}}]^T,$$

$$\boldsymbol{\nu}(k) = [\nu_{t_{R,1}}, \nu_{t_{F,1}}, \nu_{t_{R,2}}, \nu_{t_{F,2}}]^T;$$

and the optimal IFS vector at the $(k+1)$–th macro–period as:

$$\boldsymbol{v}(k) = [v_{M1}(k), v_{M2}(k), v_{M3}(k), v^1_{M4}(k), v^2_{M4}(k), v_{M4}(k), v_{M5}(k), v_{B1,M3}(k), v_{B3,M3}(k)]^T.$$

The optimal control policy we use in this example consists in the maximization of the system throughput (the IFS v_{M_5}) while minimizing the number of input parts. More precisely, at each macro–period we solve an optimization problem of the form

$$\max_{\boldsymbol{v} \in \mathcal{S}_k} J = 1000 \cdot v_{M5}(k) - (v_{M1}(k) + v^2_{M4}(k)) \tag{21}$$

where the different weighting coefficients have been chosen so as to consider the throughput maximization as the prior requirement, and the minimization of the flow of row parts taken from buffer B_0 as a secondary requirement.

k=0. Let the initial time be $\tau_0 = 0$ and the initial marking be that represented in figure 9. All timers have initially a zero value. Thus the state vector at the beginning of the first macro–period is

$$\boldsymbol{x}(0) = [10^3,\ 10^3,\ 0,\ 50,\ 0,\ 50,\ 0,\ 40,\ 0,\ 0,\ 0,\ 30,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0]^T.$$

The enabled timed transitions are $t_{11} = t_{F,1}$ and $t_{13} = t_{F,2}$, so we extract their timer set–points from the corresponding distribution: assume $\hat{\nu}_{11} = 16$ and $\hat{\nu}_{13} = 5$.

The optimal IFS vector is computed by solving an optimization problem of the form (21) where

$$\mathcal{S}_0 \ : \ \begin{cases} v_{M1} & \leq \ V_{M1} & (1) \\ v_{M2} & \leq \ V_{M2} & (2) \\ v_{M3} & \leq \ V_{M3} & (3) \\ v_{M4} & \leq \ V_{M4} & (4) \\ v_{M5} & \leq \ V_{M5} & (5) \\ v_{M3} & = \ v_{B1,M3} + v_{B3,M3} & (6) \\ v_{M4} & = \ v^1_{M4} + v^2_{M4} & (7) \\ v^1_{M4} + v_{B1,M3} & \leq \ v_{M1} & (8) \\ v_{M2} & \leq \ \alpha_2 V_{M3} & (9) \\ v_{B3,M3} & \leq \ v_{M2} & (10) \\ v_{M5} & \leq \ v^1_{M4} + (1-\alpha_2) v_{M3} & (11) \\ v_{M5} & \leq \ v^2_{M4} & (12) \end{cases} \qquad (22)$$

The optimal IFS vector is $\boldsymbol{v}(0) = [10, 15, 23.53, 1.47, 8.53, 10, 8.53, 8.53, 15]^T$.

We compute

$$\Psi_0 = \{(\pi_1, \Delta_{\pi_1}), (\pi_2, \Delta_{\pi_2}), (\pi_4, \Delta_{\pi_4}), (\pi_6, \Delta_{\pi_6}), (\pi_8, \Delta_{\pi_8}), (\pi_{10}, \Delta_{\pi_{10}}), (\pi_{12}, \Delta_{\pi_{12}}), (\gamma_{11}, \Delta_{\gamma_{11}}), (\gamma_{13}, \Delta_{\gamma_{13}})\},$$

where $\Delta_{\pi_1} = \Delta_{\pi_2} = 100$, $\Delta_{\pi_4} = \Delta_{\pi_8} = \Delta_{\pi_{10}} = \Delta_{\pi_{12}} = \infty$, $\Delta_{\pi_6} = 40.32$, $\Delta_{\gamma_{11}} = 16$ and $\Delta_{\gamma_{13}} = 5$.

Thus γ_{13}, i.e. the firing of discrete transition $t_{13} = t_{F,2}$, is the next event to occur: this corresponds to the failure of machine M_2. The length of the first macro–period is $\Delta(1) = \Delta_{\gamma_{13}} = 5$ and $\tau_1 = \Delta_{\gamma_{13}}$.

k=1. The state vector at the beginning of the second macro–period is

$$\boldsymbol{x}(1) = [995,\ 995.74,\ 0,\ 50,\ 7.36,\ 42.65,\ 0,\ 40,\ 0,\ 30,\ 0,\ 30,\ 42.65,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1,\ 0,\ 5,\ 0,\ 0]^T,$$

i.e. all machines — excepted M_2 — are operational and all finite capacity buffers — excepted B_2 — are empty.

The enabled timed transitions are $t_{11} = t_{F,1}$ and $t_{12} = t_{R,2}$. The timer set point of transition t_{11} keeps the previous value, while a timer set–point needs to be extracted for transition t_{12}: assume $\hat{\nu}_{12} = 14$.

The set of admissible IFS vectors is now defined by the new constraint set $\mathcal{S}_1$, whose only differences with respect to $\mathcal{S}_0$ are in constraints (2) and (9), both replaced by the single equation

$$v_{M2} = 0$$

due to the failure of machine M_2.

The optimal IFS vector is $\boldsymbol{v}(1) = [10,\ 0,\ 5.88,\ 4.12,\ 5.88,\ 10,\ 5.88,\ 5.88,\ 0]^T$, i.e. the failure of machine M_2 forces the assembly machine M_5 to work at a lesser rate than its MFS.

We compute

$$\Psi_1 = \{(\pi_1, \Delta_{\pi_1}), (\pi_2, \Delta_{\pi_2}), (\pi_4, \Delta_{\pi_4}), (\pi_6, \Delta_{\pi_6}), (\pi_8, \Delta_{\pi_8}), (\pi_{10}, \Delta_{\pi_{10}}), (\pi_{12}, \Delta_{\pi_{12}}), (\gamma_{11}, \Delta_{\gamma_{11}}), (\gamma_{12}, \Delta_{\gamma_{12}})\},$$

where $\Delta_{\pi_1} = 99.5$, $\Delta_{\pi_2} = 169.34$, $\Delta_{\pi_4} = \Delta_{\pi_8} = \Delta_{\pi_{10}} = \Delta_{\pi_{12}} = \infty$, $\Delta_{\pi_6} = 7.928$, $\Delta_{\gamma_{11}} = 11$ and $\Delta_{\gamma_{12}} = 14$.

Then π_6, i.e. the emptying of the continuous place $\bar{p}_{B2} = p_6$, is the next event to occur: this corresponds to the filling of buffer B_2. The length of the second macro–period is $\Delta(2) = \Delta_6 = 7.928$ and $\tau_2 = \tau_1 + \Delta_6 = 12.928$.

k=2. At the beginning of this macro–period the state vector is equal to

$$\boldsymbol{x}(2) = [987.72,\ 991.71,\ 0,\ 50,\ 50,\ 0,\ 0,\ 40,\ 0,\ 0,\ 0,\ 30,\ 89.28,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1,\ 0, 12.928, 7.928, 0]^T,$$

i.e. all machines — excepted M_2 — are operational; buffer B_2 is full, and all other finite capacity buffers are empty.

The timer set points keep the previous values, because no timed transition changes its enabling state.

The set of admissible IFS vectors is now defined by the new constraint set $\mathcal{S}_2$, that is obtained from $\mathcal{S}_0$ by replacing constraints (2) and (9) with

$$\begin{cases} v_{M2} & = & 0 & (2'') \\ \alpha_2 v_{M3} & \leq & v_{M2} & (9'') \end{cases}$$

since machine M_2 is down and buffer B_2 is full (i.e. place $\overline{p}_{B2}$ is empty).

The optimal IFS vector is $\boldsymbol{v}(2) = [5,\ 0,\ 0,\ 5,\ 5,\ 10,\ 5,\ 0,\ 0]^T$. Note that, machine M_3 cannot work, since a fixed ratio of its processed parts should have been put into buffer B_2, but this is not possible now, being buffer B_2 full.

We compute

$$\Psi_2 = \{(\pi_1, \Delta_{\pi_1}), (\pi_2, \Delta_{\pi_2}), (\pi_4, \Delta_{\pi_4}), (\pi_5, \Delta_{\pi_5}), (\pi_8, \Delta_{\pi_8}), (\pi_{10}, \Delta_{\pi_{10}}), (\pi_{12}, \Delta_{\pi_{12}}), (\gamma_{11}, \Delta_{\gamma_{11}}), (\gamma_{12}, \Delta_{\gamma_{12}})\},$$

where $\Delta_{\pi_1} = 197.41$, $\Delta_{\pi_2} = \Delta_{\pi_4} = \Delta_{\pi_5} = \Delta_{\pi_8} = \Delta_{\pi_{10}} = \Delta_{\pi_{12}} = \infty$, $\Delta_{\gamma_{11}} = 3.072$ and $\Delta_{\gamma_{12}} = 6.072$.

Thus γ_{11}, i.e. the firing of transition $t_{F,1} = t_{11}$, is the next event to occur: this corresponds to the failure of machine M_1. The length of the third macro–period is $\Delta(3) = \Delta_{\gamma_{11}} = 3.072$ and $\tau_3 = \tau_2 + \Delta_{\gamma_{11}}$.

k=3. At the beginning of the fourth macro–period the state vector is

$$\boldsymbol{x}(3) = [985.54,\ 989.54,\ 0,\ 50,\ 50,\ 0,\ 0,\ 40,\ 0,\ 0,\ 0,\ 30,\ 135.91,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 1,\ 0,\ 0,\ 11,\ 0]^T,$$

i.e. all machines — excepted M_1 and M_2 — are operational; buffer B_2 is full while all other finite capacity buffers are empty.

The enabled timed transitions are $t_{10} = t_{R,1}$ and $t_{12} = t_{R,2}$. The timer set point of transition t_{12} keeps the previous value, while a timer set–point needs to be extracted for transition t_{10}: assume $\hat{\nu}_{10} = 10$.

The set of admissible IFS vectors is now defined by the new constraint set $\mathcal{S}_3$, that is obtained from $\mathcal{S}_0$ by replacing constraints (1), (2) and (9) with the following constraints

$$\begin{cases} v_{M1} & = & 0 & (1'') \\ v_{M2} & = & 0 & (2'') \\ \alpha_2 v_{M3} & \leq & v_{M2} & (9'') \end{cases}$$

since machine M_1 and M_2 are down and buffer B_2 is full (i.e. place $\overline{p}_{B2}$ is empty).

The optimal IFS vector is $\boldsymbol{v}(3) = [0, 0, 0, 0, 0, 0, 0, 0, 0]^T$, that means that the whole production system is completely blocked during this macro–period. This can be easily understood by taking into account that, being M_5 an assembly station that contemporary takes parts of class 1 and 2, it cannot work if the flow of parts of class 1 is interrupted.

We compute

$$\Psi_3 = \{(\pi_1, \Delta_{\pi_1}), (\pi_2, \Delta_{\pi_2}), (\pi_4, \Delta_{\pi_4}), (\pi_5, \Delta_{\pi_5}), (\pi_8, \Delta_{\pi_8}), (\pi_{10}, \Delta_{\pi_{10}}), (\pi_{12}, \Delta_{\pi_{12}}), (\gamma_{10}, \Delta_{\gamma_{10}}), (\gamma_{12}, \Delta_{\gamma_{12}})\},$$

where $\Delta_{\pi_1} = \Delta_{\pi_2} = \Delta_{\pi_4} = \Delta_{\pi_5} = \Delta_{\pi_8} = \Delta_{\pi_{10}} = \Delta_{\pi_{12}} = \infty$, $\Delta_{\gamma_{10}} = 10$ and $\Delta_{\gamma_{12}} = 3$.

Thus, γ_{12} is the next event to occur: this corresponds to the repairing of machine M_2. The length of this macro–period is $\Delta(4) = \Delta_{\gamma_{12}} = 3$ and $\tau_4 = \tau_3 + \Delta_{\gamma_{12}} = 19$.

k=4. At the beginning of the fifth macro–period the state vector is

$$\boldsymbol{x}(4) = [985.54, 989.54, 0, 50, 50, 0, 0, 40, 0, 0, 0, 30, 135.91, 0, 0, 0, 0, 0, 1, 1, 0, 4, 0, 0, 0]^T.$$

The continuous marking $\boldsymbol{m}^c(4)$ is equal to $\boldsymbol{m}^c(3)$, since no continuous transition has fired during the previous macro–period.

The set of admissible IFS vectors is now defined by the new constraint set $\mathcal{S}_4$, that is obtained from $\mathcal{S}_0$ by replacing constraints (1) and (9) with

$$\begin{cases} v_{M1} & = & 0 & (1'''') \\ \alpha_2 v_{M3} & \leq & v_{M2} & (9'''') \end{cases}$$

since machine M_1 is down and buffer B_2 is full (i.e. place $\overline{p}_{B2}$ is empty).

The optimal IFS vector is $\boldsymbol{v}(4) = [0, 15, 15, 0, 4.5, 4.5, 4.5, 0, 15]^T$.

This evolution can also be described graphically as shown in figure 10, representing the most significant state components: figures 10.(a)$\cdots$(d) show the markings of the continuous places corresponding to non empty buffers; figures 10.(d)$\cdots$(h) show the markings of discrete places. The symbol 'o' has been used to represent the discrete numerical values computed in accordance to the linear state space model. Within each macro–period the linearity of the curves is due to the first–order approximation.

[Insert figure 10 about here]

7 Conclusions

In this paper we have shown that First–Order Hybrid Petri Nets are well suited to represent manufacturing systems, providing a unifying description of the discrete states of a system and of the corresponding approximated fluid models.

The elementary subnets describing different components of a manufacturing system can be composed in a bottom-up fashion to derive the overall model. The choice of the optimal machine production rates corresponds to determining the instantaneous firing speed vector of continuous transitions. This has been solved myopically using a linear programming approach.

A linear time-varying discrete-time state variable model has been used to describe the system behaviour, thus providing the basis of an efficient simulation tool. There are two main issues that have not been addressed in this paper and may be the object of future research.

Firstly, we plan to address the problem of performance optimization of manufacturing systems. There are two aspects in this issue: optimization of the system performances by a suitable choice of the IFS of continuous transitions, and optimization of the system design parameters, such as buffer capacity, maximal machine production rates, machine reliability, etc. In (Balduzzi et al. 2000) these aspects have been partially addressed within a macro–period using the sensitivity analysis methods that pertain to First–Order Hybrid Petri Nets. We feel that it is necessary to also address them within a given horizon that may span over several macro–periods.

Secondly, it is important to characterize relevant properties of manufacturing systems such as stability, bottlenecks, deadlock, etc. These properties should be related to properties of FOHPN and suitable algorithms should be derived to validate them. As an example, stability may be related to place boundedness or absence of Zeno executions (i.e. in an given finite horizon only a finite number of macro–events may occur); deadlock may be related to liveness, etc.

References

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

Alla, H., David, R., 1998, Continuous and Hybrid Petri Nets, *Journal of Circuits, Systems and Computers*, **8**, 159–188.

Alla, H., David, R., 1998, A Modelling and Analysis Tool for Discrete Event Systems: Continuous Petri Net, *Performance Evaluation*, **33**, 175–199.

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

Balduzzi, F., Menga, G., 1998, A State Variable Model for the Fluid Approximation of Flexible Manufacturing Systems, Proceedings IEEE Int. Conf. Robotics and Automation, Leuven, Belgium, pp. 1172–1778.

Balduzzi, F., Giua, A., Menga, G., 2000, First–Order Hybrid Petri Nets: a Model for Optimization and Control, *IEEE Trans. on Robotics and Automation*, **16**.

Balduzzi, F., Giua, A., Menga, G., Seatzu, C., 1999, A Linear State Variable Model for First–Order Hybrid Petri Nets, Proceedings of the 14–th IFAC World Congress, Bejing, P.R. China, pp. 205–210.

Buzacott, J.A., Hanifin, L.E., 1978, Models of automatic transfer lines with inventory banks: A review and comparison, *AIIE Trans.*, **10**, 197–207.

Caramanis, M., 1987, Production System Design: a Discrete Event Dynamic System and Generalized Benders' Decomposition Approach, *International Journal of Production Research*, **25**, 1223–1234.

David, R., 1997, Modeling of Hybrid Systems Using Continuous and Hybrid Petri Nets, Proceedings of the 7–th Petri Nets and Performance Models, Saint Malo, France, pp. 47–58.

David, R., Alla, H., 2001, Réseaux de Petri Hybrides, in *Modélisation et commande des systèmes dynamiques hybrides*, Hermés Ed. Paris, France.

Demongodin, I., Koussoulas, N.T., 1998, Differential Petri Nets: Representing Continuous Systems in a Discrete–Event World, *IEEE Transactions on Automatic Control*, **43**, 573–579.

Murata, T., 1989, Petri Nets: Properties, Analysis and Applications, *Proceedings IEEE*, **77**, 541–580.

Phillis, Y.A., Kouikoglou, V.S., 1994, Discrete Event Modeling and Optimization of Unreliable Production Lines with Random Rates, *IEEE Trans. Robotics and Automation*, **10**, 153–159.

Sethi, S.P., Zhang, Q., 1994, *Hierarchical Decision Making in Stochastic Manufacturing Systems*, (Birkhäuser Boston, Cambridge, MA).

Sharifnia, A., 1994, Stability and Performance od Distributed Production Control Methods based on Continuous–Flow Models, *IEEE Trans. Automatic Control*, **39**, 725–737.

Suri, R., Fu, B.R., 1994, On Using Continuous Flow Lines to Model Discrete Production Lines, *J. Discrete Event Dynamic Systems*, 129–169.

Trivedi, K.S., Kulkarni, V.G., 1993, FSPNs: Fluid Stochastic Petri Nets, *Lecture Notes in Computer Science*, **691**, M. Ajmone Marsan (ed.), Proc. 14th Int. Conf. on Applications and Theory of Petri Nets, (Springer-Verlag, Heidelberg), 24–31.

List of figures captions