HYBRID STOCHASTIC PETRI NETS: FIRING SPEED COMPUTATION AND FMS MODELLING

Fabio Balduzzi[1] Alessandro Giua[2] Giuseppe Menga[1]

[1] Dip. di Automatica ed Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy. Phone: +39-011-564.7025; Fax: +39-011-564.7099; Email: (menga, balduzzi)@polito.it.
[2] DIEE, Università di Cagliari, Piazza d'Armi, 09123 Cagliari, Italy. Phone: +39-070-675.5892; Fax: +39-070-675.5900; Email: giua@diee.unica.it.

Keywords

Hybrid Petri nets, flexible manufacturing systems, linear programming.

Abstract

In this paper we adopt the fluid approximation theory to describe the dynamic behavior of Flexible Manufacturing Systems that we model with Hybrid Stochastic Petri Nets, a class of nets in which some places may hold fluid rather than discrete tokens. The continuous transitions of the net are fired with speeds that are piecewise constants over the entire time horizon and their instantaneous values can be obtained by solving a sequence of linear programming problems. Conflicts among continuous transitions correspond to scheduling decisions, and we discuss several optimization schemes that can be used to resolve them.

1 Introduction

We consider *Flexible Manufacturing Systems* (FMS) consisting of a set of a stations with unreliable machines and buffers of finite capacity, among which several parts of a certain class are circulated and processed. To describe the dynamic behavior of such systems we adopt the Petri Net (PN) formalism. Since in practical problems the number of reachable states may explode, we develop a hybrid (discrete–event and continuous–flow) model.

Fluid Stochastic Petri Nets have been introduced by Kulkarni and Trivedi in [8] in order to extend the stochastic Petri nets framework of [1]. They proposed a model with places holding continuous tokens and arcs representing fluid flows, defining rules for transitions enabling and firing. In this paper we define a *hybrid* model of the net in which places and transitions may be either continuous or discrete, following the hybrid framework introduced by Alla and David in [2], and we allow fluids to move smoothly through the net. Hybrid Petri nets whose continuous places may contain negative real tokens have also been defined in the literature (e.g., [7]) but will not be considered here.

In the *Hybrid Stochastic Petri Net* framework (HSPN), a net consists of continuous places holding fluid, discrete places containing a non–negative integer number of tokens and transitions, either discrete or continuous. Enabled continuous and discrete transitions may then fire according to their firing speeds or time delays, respectively. We describe the dynamics of an HSPN by setting up a linear discrete–time state variable model. Thus hybrid Petri nets allows us to model manufacturing systems by means of first–order fluid approximations, where the marking of continuous places are piecewise linear and continuous functions of time.

The main motivation of this paper is to put modelling issues encountered when dealing with manufacturing systems in the context of hybrid stochastic Petri nets. Precisely we propose a neat formulation of the fluid model which describes the evolution in time of an FMS that is driven by the occurrence of a limited number of events, that we call *macro–events*. Then the system evolves through a sequence of *macro–states* characterized by the functional status of each service.

Conflict resolution is an important issue in the study of (discrete) timed nets. We have a conflict when a limited number of tokens enables more than one transition but it is only sufficient to fire a subset of them. Several schemes have been devised to tackle this problem, including *token reservations* [2], *re-sampling rules* and *priorities* [1]. In the present work, we use hybrid nets to model FMSs, and conflicts arise at continuous places, where production flows must be routed in the system. The conflict resolution policy represents the decision that a plant operator must take in order to optimize the process. This decision may be based on local or global information and requires computing the instantaneous firing speeds of continuous transitions. We will provide a formal description for the calculation of the instantaneous firing speeds of the continuous transitions, obtained by solving a linear programming problem. The different objective functions of this optimization problem correspond to different policies.

Briefly, the rest of the paper is structured as follows. In section 2 we introduce the Petri net formalism used in the following sections and we develop the hybrid model of stochastic Petri nets. In Section 3 we show how HSPNs can be used to derive a first–order fluid model of an FMS. Section 4 introduces the concepts of macro–states and macro–events. Section 5 is concerned with the computation of the instantaneous firing speed of continuous transitions and with different conflict resolution schemes.

2 Background

We recall the Petri net formalism used in this paper. For a more comprehensive introduction to place/transition Petri nets see [10], while the common notation and semantics for GSPNs can be found in [1]. The first approach towards continuous Petri nets was carried out by Alla and David and then extended to hybrid nets in [2]. The HSPN

© 1998 IEE, WODES98 – Proc. of the Fourth Workshop on Discrete Event Systems (Cagliari, Italy), pp. 432–438.

model we use follows [2, 1]. Another class of hybrid stochastic Petri nets was also defined in [8].

An HSPN is a structure $N = (P, T, Pre, Post, \mathcal{F})$. 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 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).

$$Pre : \begin{cases} P_d \times T \to \mathbb{N} \\ P_c \times T \to \mathbb{R}^+ \cup \{0\} \end{cases}$$

and

$$Post : \begin{cases} P_d \times T \to \mathbb{N} \\ P_c \times T \to \mathbb{R}^+ \cup \{0\} \end{cases}$$

are the *pre-* and *post-incidence functions* that specify the arcs. We require (*well-formed nets*) that for all $t \in T_c$ and for all $p \in P_d$, $Pre(p, t) = Post(p, t)$. The function $\mathcal{F}$ is defined for continuous and discrete timed transitions so that $\mathcal{F} : T \setminus T_I \to \mathbb{R}^+$. We associate to a continuous transition $t_i \in T_c$ its *maximum firing speed* (MFS) $V_i = \mathcal{F}(t_i)$. We associate to a deterministic timed transition $t_i \in T_D$ its (constant) firing delay $\delta_i = \mathcal{F}(t_i)$. We associate to an exponentially distributed timed transition $t_i \in T_E$ its average firing rate $\lambda_i = \mathcal{F}(t_i)$, i.e. the average firing delay is $\frac{1}{\lambda_i}$, where λ_i is the parameter of the corresponding exponential distribution.

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 $\mathbf{C}(p, t) = Post(p, t) - Pre(p, t)$. The restriction of $\mathbf{C}$ to P_X and T_Y ($X, Y \in \{c, d\}$) is denoted $\mathbf{C}_{XY}$. Note that by the well-formedness hypothesis $\mathbf{C}_{dc} = 0$.

A *marking*

$$m : \begin{cases} P_d \to \mathbb{N} \\ P_c \to \mathbb{R}^+ \cup \{0\} \end{cases}$$

is a function that assigns to each discrete place a nonnegative number of tokens, represented by black dots and assigns to each continuous place a fluid volume; m_p denotes the marking of place p. The value of a marking at time τ is denoted $m(\tau)$. The restriction of m to P_d and P_c are denoted with m^d and m^c, respectively. An *HSPN system* $\langle N, m(0) \rangle$ is a HSPN N with an initial marking $m(0)$.

A discrete transition t is enabled at m if for all $p \in {}^\bullet t$, $m_p \geq Pre(p, t)$. An enabled discrete transition t fires (after the associated delay) yielding the marking $m' = m + \mathbf{C}(\cdot, t)$.

A continuous transition t is enabled at m if for all $p \in {}^{(d)}t$, $m_p \geq Pre(p, t)$. Note that the enabling of a continuous transition does not depend on the marking of its continuous input places. We distinguish *strongly enabled* and *weakly enabled* continuous transitions. A transition $t_i \in T_c$ is strongly enabled at $m(\tau)$ if for all places $p \in {}^{(c)}t$, $m_p(\tau) > 0$. Then it may fire with an *instantaneous firing speed* (IFS) $v_i(\tau) = V_i$. A transition $t_i \in T_c$ is weakly enabled at $m(\tau)$ if for some $\bar{p} \in {}^{(c)}t$, $m_{\bar{p}}(\tau) = 0$. Thus its IFS may result $v_i(\tau) < V_i$ because it cannot remove more fluid from place $\bar{p}$ than the quantity entered in $\bar{p}$ by other transitions. Moreover if $t_i \in T_c$ is not enabled at $m_p(\tau)$ then $v_i(\tau) = 0$.

Figure 1: Firing of a continuous transition.

We can now define the macro–behavior of a net. A *macro–event* occurs when: (a) either a discrete transition fires, thus changing the discrete marking and enabling/disabling a continuous transition; (b) or a continuous place becomes empty, thus changing the enabling state of a continuous transition from strong to weak. Let τ_k and τ_{k+1} be the occurrence in time of consecutive macro–events; the interval of time $\Delta_k = [\tau_k, \tau_{k+1})$ is called a *macro–period*. We will assume that the IFS of continuous transitions are piecewise constant during a macro–period. Thus the discrete marking and the IFS vector during a macro–period define a *macro–state* that correspond to the *invariant behavior states* of [2]. These settings are illustrated in the example below.

Example 1. Transitions t_1 and t_2 in Figure 1 have associated MFSs V_1 and V_2. We assume $V_1 \cdot a < V_2 \cdot b$ and $m_p(0) = l_p > 0$. As long as p is not empty transitions t_1 and t_2 fire at their maximum speed, that is $v_1(\tau) = V_1$ and $v_2(\tau) = V_2$. The marking of p is given by:

$$\begin{cases} m_p(0) = l_p \\ \frac{\mathrm{d}m_p}{\mathrm{d}\tau} = v_1 \cdot b - v_2 \cdot a \end{cases}$$

At time $\tau_e = \frac{l_p}{v_2 \cdot b - v_1 \cdot a}$, t_2 cannot fire at its maximum speed because p is empty. Hence for $\tau > \tau_e$ transition t_2 is weakly enabled with IFS $v_2(\tau) = V_1 \cdot \frac{a}{b}$. The event "place p becomes empty" at time τ_e has modified the evolution of the system changing the IFS of the continuous transitions. ■

Let $v_i(\tau)$ be the IFS of each transition $t_i \in T_c$. We can write the equation which governs the evolution in time of the marking of a place $p \in P_c$ as

$$\frac{\mathrm{d}m_p(\tau)}{\mathrm{d}\tau} = \sum_{t_i \in T_c} \mathbf{C}(p, t_i) \cdot v_i(\tau) \qquad (1)$$

Indeed Equation 1 holds assuming that at time τ no discrete transition is fired and that all speeds $v_i(\tau)$ are continuous in τ. The evolution in time of the marking of a place $p \in P_d$ is governed by the common enabling and firing rules defined in [1].

We now describe the dynamics of an HSPN by setting up a linear discrete–time state variable model. Let τ_0 be the initial time, τ_k $(k > 0)$ be the instants in which macro–events occur, $\mathbf{v}(\tau_k)$ be the IFS vector during the macro–period Δ_k and $\sigma(\tau_k)$ the firing count vector at time τ_k. Then the continuous behavior of an HSPN can be described during a macro–period Δ_k by

$$\begin{cases} m^c(\tau) & = m^c(\tau_k) + \mathbf{C}_{cc} \cdot \mathbf{v}(\tau_k) \cdot (\tau - \tau_k) \\ m^d(\tau) & = m^d(\tau_k) \end{cases} \tag{2}$$

where $\tau \in [\tau_k, \tau_{k+1})$, while its discrete behavior at the occurrence of macro–events is described by

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

3 Description of the FMS Model

Hybrid Petri nets allows us to model manufacturing systems with first–order fluid approximations. Indeed fluid models are well studied and documented in the literature, and the readers are referred to Chen and Mandelbaum [9] for references on the *fluid approximation theory*. Specifically we consider an FMS consisting of a set of n single–server stations among which a certain class of continuous flows (fluid) is circulated and processed. A more general FMS configuration with different classes of products flowing through has been deeply studied by Balduzzi and Menga in [3] by using first and second order fluid approximations.

Stations are denoted by M_i, for $i = 1, \ldots, n$, and are represented in the HSPN as continuous transitions $t_{M_i} \in T_c$ which firing corresponds to a continuous production at rate $v_{M_i}(\tau)$ when the input buffer B_i is not empty. Working stations (generically called services) are represented within the FMS configuration by unreliable machines coupled with input buffers of finite capacity. Each machine has its own input buffer to accommodate the inflow of parts. Since machines are unreliable we may consider randomly occurring failures as operation–dependent or time–dependent (readers are referred to Buzacott [5]). An operation–dependent failure can occur only when the machine is working at a given production rate and as suggested in [5] this model is more appropriate than the equivalent time–dependent model when dealing with manufacturing systems. However if we are interested in evaluating performance measures assuming time–dependent failures, each service can be appropriately modelled as an HSPN. The considered control scheme is shown in Figure 2. This is the same model presented in [2] for a an unreliable machine producing at a constant rate V_{M_i}.

Transition t_{M_i} models the production of machine M_i at a rate given by its IFS v_{M_i}. The maximum machine production rate is defined by its MFS V_{M_i}. Continuous firing of transition t_{M_i} corresponds to a continuous production at rate $v_{M_i} \leq V_{M_i}$ when the input buffer $p_{B,i}$ is not empty. Obviously the machine will keep on producing only if it is *operational*, that is the place $p_{up,i}$ (*Machine Up*) is marked. When the machine breaks down, independently on the production volume currently processed, $p_{up,i}$ is not marked and $p_{down,i}$ is marked, hence t_{M_i} is not enabled, then not fired.

Figure 2: *Time–Dependent* failures model.

Figure 3: *Operation–Dependent* failures model.

On the other hand assuming operation–dependent failures we model each service as shown in Figure 3. Continuous places $p_{R,i}$ and $p_{F,i}$ hold fluids represented by the production volume w_i that will be processed by the machine M_i before breaking. This value can be either deterministic or obtained as a sample drown from an i.i.d random variable. Machine M_i breaks down after processing the fluid quantity w_i at a production rate defined by the IFS of transition t_{M_i}. At that time the continuous place $p_{F,i}$ will be filled up by the fluid volume w_i, thus enabling the immediate transition $t_{down,i}$. Firing of transition $t_{down,i}$ consists in taking out the volume w_i from $p_{F,i}$ and adding 1 token to the discrete place $p_{down,i}$, which is representing the condition of *Machine Down* for this service. Hence the *failure* event occurred at machine M_i has made a change on the state of this service and, as a consequence, the IFS of transition t_{M_i} will be $v_{M_i} = 0$. Then the machine will be under repairing as long as the repair event does occur. That is, after the interval of time $\frac{1}{\lambda_{R,i}}$, representing the delay after which the discrete transition $t_{R,i}$ will be fired, the service gets repaired and place $p_{up,i}$ will be marked, thus representing the condition of *Machine Up* for this service. At the same time the immediate transition $t_{up,i}$ is enabled and it can get fired providing an *impulsive signal* to the continuous place $p_{R,i}$ which will then filled up by the fluid volume w_i. An impulsive signal is here informally employed to provide continuous places with initial conditions, thus representing the loading of reservoirs of fluid, i.e. place $p_{R,i}$, at the production volume that will be processed by the machine before the next failure. This model highlights the transformation of fluids into discrete tokens and

Figure 4: The output model of a machine.

Figure 5: The model of a finite buffer.

vice versa through discrete transitions.

In Figure 4 we have depicted the output model of a machine. Parts move generically from station M_i to the input buffer B_j of machine M_j according to their production cycle. Continuous place p_{disp} acts as a dispatcher of capacity equal to 0 for the flow of parts at the output of each machine. Let $v_{M_i,B_j}(\tau)$ by the outflow rate of parts moving from machine M_i to B_j. Then this structure allows us to model the routing of parts within the net ensuring that $\sum_{t_{M_i,B_j} \in T_c} v_{M_i,B_j}(\tau) \leq v_{M_i}(\tau)$, for $\tau > 0$. Note that we do not need to bound the IFS of the outflow continuous transitions t_{M_i,B_j}. Along their route, parts are queued in buffers, one for each machine, which are represented in the HSPN by continuous places $p_{B_i} \in P_c$ with bounded capacity c_i. This condition is represented by the co-buffer $p_{\bar{B}_i}$ such that $m_{\bar{B}_i}(0) = c_i$, as shown in Figure 5.

We denote with $\mathbf{c} = [c_1, \ldots, c_n]^T$ the buffer capacity vector and, assuming operation–dependent failures (readers are referred to Figure 3), we define for each machine the *Production Volume Before Breaking*, denoted by w_i, and the *Repairing Time*, denoted by $d_i = \frac{1}{\lambda_{R,i}}$, both assumed as independent identically distributed random variables. Machine service times are indicated with η_i and are also assumed independent random variables with identical distribution. Hence the MFS of the continuous transitions are defined as $V_{M_i} = \mathcal{F}(t_{M_i}) = \frac{1}{E[\eta_i]}$.

4 Macro–behavior of the FMS

The evolution in time of the first–order fluid approximation of an FMS modelled with hybrid stochastic Petri nets is driven by the occurrence of a limited number of events (machine starvation, blockage, breakdown and repair) that in our framework are called *macro–events*. Then the FMS evolves through a sequence of macro–states, characterized by the functional status of the physical components of all services: the machines, *operational* or *down*, and the buffers, *full*, *not full-not empty*, *empty*.

In this paper we consider *linear* fluid models assuming that input and output processes corresponding to the inflow and outflow of parts at each machine are linear functions of time. Let τ_k, for $k = 1, 2, \ldots$, be points in time corresponding to the occurrence of the macro–events, and $v_i(\tau_k)$ the IFS of transition $t_i \in T_c$ during the macro–period Δ_k. For each place $p_{B_i} \in P_c$ the total inflow and outflow rate of fluids is

$$\begin{aligned} \phi_{i,in}(\tau_k) &= \textstyle\sum_h v_{M_h,B_i}(\tau_k) \\ \phi_{i,out}(\tau_k) &= v_{M_i} = \textstyle\sum_h v_{M_i,B_h}(\tau_k) \end{aligned}$$

with the notation of Figures 4 and 5.

The macro–state of the system does change whenever discrete transitions fire and(or) continuous transitions have been modified their enabling conditions as a consequence of certain macro–events occurred at their input places. We denote the set of macro–events by $\mathcal{E} = \{F_i, R_i, BE_i, BF_i\}$ which elements are defined as follows:

- $\mathbf{F_i}$ (*Failure of machine* M_i). After a machine is repaired, failures will occur after the production volume $\tilde{w}_i$.
- $\mathbf{R_i}$ (*Repair of machine* M_i). When a machine fails, it will be repaired after $\tilde{d}_i$ time units.
- $\mathbf{BF_i}$ (*Buffer Full at machine* M_i). The buffer level reaches its capacity c_i while $\phi_{i,in}(\tau) \geq \phi_{i,out}(\tau)$.
- $\mathbf{BE_i}$ (*Buffer Empty at machine* M_i). The buffer level reaches 0 while $\phi_{i,in}(\tau) \leq \phi_{i,out}(\tau)$.

The macro–event set $\mathcal{E}$ has been defined with regard to the *physical* events which usually do occur in an FMS. In the HSPN framework those macro–events correspond to the firing of immediate transitions and to the adjustments made on the speeds of continuous transitions.

Following the above notation, we can characterize the admissible macro–states as follow:

- *Machine Operational*. Machine M_i reaches this macro–state at the occurrence of macro–events $\mathrm{R_i}$ and then leaves at the occurrence of macro–events $\mathrm{F_i}$.
- *Machine Broken*. Machine M_i reaches this macro–state at the occurrence of macro–events $\mathrm{F_i}$ and then leaves it at the occurrence of macro–events $\mathrm{R_i}$. The IFS $v_i(\tau)$ goes to 0 because transition t_{M_i} is disabled.
- *Buffer Full*. Macro–state reached at the occurrence of macro–events $\mathrm{BF_i}$. Services leave it at the occurrence of any macro–event which will result in the condition $\phi_{i,in}(\tau) \leq \phi_{i,out}(\tau)$.
- *Buffer Empty*. Macro–state reached at the occurrence of macro–events $\mathrm{BE_i}$. Services leave it at the occurrence of any macro–event which will result in the condition $\phi_{i,in}(\tau) \geq \phi_{i,out}(\tau)$.

- *Buffer not-Full not-Empty*. In any other macro–state reached at the occurrence of exogenous macro–events services will be considered operating under the heavy traffic conditions.

5 Firing speed and dynamics of an HSPN

The computation of an admissible IFS vector of continuous and hybrid nets is not trivial. In [6] an iterative algorithm was given to determine one admissible vector; the algorithm aims at maximizing speeds while respecting priority rules. We propose a different approach, in which we use linear inequalities to define the set of all admissible firing speed vectors $\mathcal{S}$. Each vector $\mathbf{v} \in \mathcal{S}$ 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. There are several advantages in our approach.

- We can explicitly define the set of all admissible IFS vectors in a given macro-state and not just compute a particular vector.
- We consider more general scheduling rules than priorities. In general in an FMS we may want to: maximize machines utilization, maximize the throughput of the system, balance the load, etc. Each of these problems corresponds to a particular objective function. Note that each set $\mathcal{S}$ corresponds to a particular system macro-state. Thus, our optimization scheme can only be *myopic* [3], in the sense that it generates a piecewise optimal solution, i.e., a solution that is optimal only in a macro–period .
- 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.
- Linear programming leads to sensitivity analysis, which plays an essential role in performance evaluation and optimization. In fact, we may be able to compute analytically the objective function improvement due to a parameter variation.

5.1 Admissible IFS vectors and conflicts

Definition 2 (admissible IFS vectors). Let N be an HSPN, with n_c continuous transitions, incidence matrix $\mathbf{C}$, and current marking m. Let $T_{\mathcal{E}}(m) \subset T_c$ ($T_{\mathcal{N}}(m) \subset T_c$) be the subset of continuous transitions enabled (not enabled) at m, while $P_{\mathcal{E}} = \{p \in P_c \mid m_p = 0\}$ is the subset of continuous places that are empty. Any *admissible IFS vector* $\mathbf{v} = [v_1, \cdots v_{n_c}]^T$ at m is a feasible solution of the following linear set:

$$\left\{ \begin{array}{lll} (a) & V_j - v_j \geq 0 & \forall t_j \in T_{\mathcal{E}}(m) \\ (b) & v_j \geq 0 & \forall t_j \in T_{\mathcal{E}}(m) \\ (c) & v_j = 0 & \forall t_j \in T_{\mathcal{N}}(m) \\ (d) & \sum_{t_j \in T_{\mathcal{E}}} \mathbf{C}(p, t_j) \cdot v_j \geq 0 & \forall p \in P_{\mathcal{E}}(m) \end{array} \right. \tag{4}$$

Thus the total number of constraints that define this set is 2card$\{T_{\mathcal{E}}(m)\}$ + card$\{T_{\mathcal{N}}(m)\}$ + card$\{P_{\mathcal{E}}(m)\}$. The set of all feasible solutions is denoted $\mathcal{S}(N, m)$. ■

Figure 6: A hybrid Petri net model of a re-entrant line.

Figure 7: A continuous place with a conflict.

Constraints of the form (4.a), (4.b), and (4.c) follow from the enabling rules. Constraints of the form (4.d) follow from (1), because if a place is empty its fluid content cannot decrease.

Example 3. Let N be the continuous net in Figure 6, with $\alpha \in (0, 1)$, where place p is initially empty. Such a net is representative of a re-entrant production line. According to the previous definition, the set $\mathcal{S}(N, m)$ is defined by the following inequalities:

$$\left\{ \begin{array}{ll} V_1 - v_1 & \geq 0 \\ V_2 - v_2 & \geq 0 \\ v_1, v_2 & \geq 0 \\ v_1 - (1-\alpha) v_2 & \geq 0 \end{array} \right. \tag{5}$$

■

We now want to use the above formalism to define the concept of *conflict* in a net. We will only consider conflicts at continuous places, an example of which is shown in Figure 7. When place p_i is not empty, both $t_{out,1}$ and $t_{out,2}$ can fire at their MFS. When place p_i is empty, however, the output flow $v_{out,1} + v_{out,2}$ is bounded by the input flow v_{in}, thus in the constraint set $\mathcal{S}(N, m)$ there will be a constraint of the form (4.d) relative to place p_i that writes $v_{in} \geq v_{out,1} + v_{out,2}$. This constraint expresses the fact that we have a limited amount of resource (the input flow) that must be shared between different processes (the output transitions). There is no conflict, instead, if each empty place $p \in P_c$ has at most one enabled output transition $t \in T_c$. This motivates next definition.

Definition 4 (continuous conflict free). Let N be an HSPN whose present macro-state is characterized by a marking m and let $\mathcal{S}(N, m)$ be the linear set defined by

(4). Any constraints of the form (4.d) can be written as

$$\sum_{j \in J} \alpha_j v_j \geq \sum_{k \in K} \alpha_k v_k \tag{6}$$

with $J \cap K = \emptyset$ and $\alpha_j, \alpha_k \in \mathbb{R}^+ \cup \{0\}$.

We say that N is *continuous conflict free* (CCF) at m if for all constraints of the form (4.d) rewritten as (6) holds card $\{K\} \leq 1$. ■

In the rest of this section, we discuss the relationship between conflict resolution (i.e., the computation of transition IFS) and performance optimization.

5.2 Conflict free firing speed computation

If we set our goal to maximize the transition firing speeds, it is possible to show that in a continuous conflict free HSPN each IFS may be maximized independently.

Theorem 5. Let N be a HSPN and m be its present marking. If N is CCF at m, the optimal solution $\mathbf{v}^*$ of the following LPP

$$\begin{array}{ll} \max & \mathbf{1}^T \cdot \mathbf{v} \\ \text{s.t.} & \mathbf{v} \in \mathcal{S}(N, m) \end{array}$$

is such that $\forall \mathbf{v} \in \mathcal{S}(N, m)$, $\mathbf{v} \leq \mathbf{v}^*$ (componentwise).

Proof. Let $\oplus$ be the (componentwise) $\max$ operator, i.e., $\mathbf{w} \oplus \mathbf{y} \equiv (w_i \oplus y_i)_i \equiv (\max\{w_i, y_i\})_i$. It is sufficient to prove that if the net is CCF, then $\mathbf{w}, \mathbf{y} \in \mathcal{S}(N, m) \Longrightarrow \mathbf{w} \oplus \mathbf{y} \in \mathcal{S}(N, m)$.

Clearly, if $\mathbf{w}$ and $\mathbf{y}$ satisfy (4), then $\mathbf{w} \oplus \mathbf{y}$ will satisfy all constraints of the form (4.a), (4.b), and (4.c). Under the hypothesis of conflict freeness, we can write any constraint of the form (4.d) associated to a place p as:

1. $\sum_{j \in J} \alpha_j v_j \geq 0$ if no enabled transition outputs from place p;

2. $\sum_{j \in J} \alpha_j v_j \geq \alpha_{out} v_{out}$ if t_{out} is the only enabled transition outputting from place p.

with $\alpha_j, \alpha_{out}, v_j, v_{out} \in \mathbb{R}^+ \cup \{0\}$.

In the first case we have that:

$$\begin{array}{l} \sum_{j \in J} \alpha_j \left(w_j \oplus y_j\right) \geq \left(\sum_{j \in J} \alpha_j w_j\right) \oplus \left(\sum_{j \in J} \alpha_j y_j\right) \\ \geq 0 \oplus 0 = 0 \end{array}$$

while in the second case we have

$$\begin{array}{l} \sum_{j \in J} \alpha_j \left(w_j \oplus y_j\right) \geq \left(\sum_{j \in J} \alpha_j w_j\right) \oplus \left(\sum_{j \in J} \alpha_j y_j\right) \\ \geq \left(\alpha_{out} w_{out}\right) \oplus \left(\alpha_{out} y_{out}\right) = \alpha_{out}\left(w_{out} \oplus y_{out}\right) \end{array}$$

i.e., the vector $\mathbf{w} \oplus \mathbf{y}$ satisfies all constraints of the form (4.d) as well. □

In the case of CCF nets, the optimal solution $\mathbf{v}^*$ in the previous theorem coincides with the solution computed with the priority algorithm in [6]. It may be interesting, however, to compare the two algorithms via an example.

Example 6. Let us consider again the net in Figure 6 whose set of admissible IFS vectors is given by (5). If we compute the vector $\mathbf{v}^*$ solution of (5) that maximizes $J = v_1 + v_2$ we clearly obtain $v_1^* = V_1$ and $v_2^* = \min\{\frac{1}{1-\alpha} V_1, V_2\}$. This example is so simple that we can write the solution in closed form; in more complex cases, the solution can still be easily found solving the associated LPP. If we apply the procedure proposed in [6], we obtain at the first iteration step $v_1 = V_1$, while to compute the IFS of transition t_2 we need to solve the following iterative problem

$$\left\{ \begin{array}{ll} v_2^0 & = 0 \\ v_2^{r+1} & = \min(V_1 + \alpha \cdot v_2^r, V_2) \end{array} \right.$$

and for $V_1 \leq (1-\alpha) V_2$ the algorithm requires an infinite number of steps to converge to the correct value $v_2 = \frac{1}{1-\alpha} V_1$. ■

5.3 Global optimization

When the net is not conflict free, not all firing speed may be maximized independently. We can always solve the conflicts, however, by solving an LPP aimed at a *global optimization* of the system resources. We may consider different performance indices as the objective function in the LP formulation of the problem. We consider some examples taken from the manufacturing domain.

1) In an FMS, the goal may be to maximize machines utilization. Thus, in a HSPN model we can consider as optimal the solution $\mathbf{v}^*$ of (4) that maximizes the performance index $J = \mathbf{1}^T \cdot \mathbf{v}$ which is of course intended to maximize the sum over all flow rates.

2) In an FMS, the goal may be to maximize throughput. Thus, in a HSPN model we may want to maximize the performance index $J = \mathbf{a}^T \cdot \mathbf{v}$ where

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

3) In an FMS, the problem of the *dynamic load balancing* consists in reducing the difference between maximum and minimum utilization of machines in a given set. In a HSPN model, the utilization of a transition t_j can be given as the ratio between v_j / V_j. Then we may want to minimize the performance index $J = \max_{j \in K}\{v_j / V_j\} - \min_{j \in K}\{v_j / V_j\}$ for a suitable index set K.

A different optimization procedure is based on *global priorities* (GP). We assume that the n_c continuous transitions of the net are ordered in a priority sequence $t_1 \succ t_2 \succ \cdots \succ t_{n_c}$. The GP-optimal solution $\mathbf{v}^* = [v_1^*, \cdots v_{n_c}^*]^T$ is such that

$$\begin{array}{ll} v_1^* & = \max\{v_1 \mid \mathbf{v} \in \mathcal{S}(N, m)\}; \\ v_2^* & = \max\{v_2 \mid \mathbf{v} \in \mathcal{S}(N, m), v_1 = v_1^*\}; \\ v_3^* & = \max\{v_2 \mid \mathbf{v} \in \mathcal{S}(N, m), v_1 = v_1^*, v_2 = v_2^*\}; \\ & \cdots \end{array}$$

This solution can be found by solving n_c LPPs. First we solve (4) with $J = v_1$ computing v_1^*; then we add to (4) the constraint $v_1 = v_1^*$ and solve with $J = v_2$; etc.

Example 7. Consider the net in Figure 8 with $V_1 = V_5 = 10$, $V_2 = V_3 = V_4 = 7$. We apply the method discussed above to obtain $\mathbf{v}^* = [10, 7, 3, 3, 10]^T$. Note that applying the algorithm proposed in [6] we obtain $\mathbf{v} = [10, 3, 7, 7, 10]^T$, that is an admissible IFS vector even though it does not have the same properties of the GP-optimal solution. ■

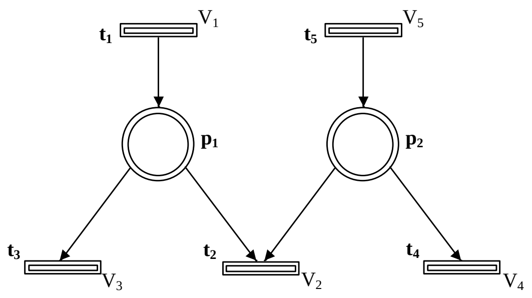

Figure 8: An HSPN with a non free-choice conflict.

Note that there exist other techniques based on lexicographic ordering [4] that may well be meaningfully used to compute the GP-optimal solution solving a single LPP with a suitably modified objective function. This will be explored in future works.

5.4 Local Optimization

The use of a performance index to be maximized (or minimized) over the space of all admissible IFS vectors, corresponds to a global optimization procedure. It is often the case, however, that local rules are used to determine the operating mode of a system described by a hybrid net. These rules correspond to decisions that can be taken in a decentralized way. We consider the case of nets where all conflicts are *free-choice*, i.e., if a continuous place p has more than one output continuous transition (e.g., $p^{(c)} = \{t_1, t_2, \cdots t_k\}$ with $k > 1$), then it is the only continuous input place for all those transitions (i.e., ${}^{(c)}t_j = \{p\},\ j = 1, \ldots, k$). The conflict in Figure 7 is free-choice, while the two conflicts in Figure 8 are not. When the conflicts are not free-choice, the local optimization rules described below may not be well founded.

One particular simple rule that may be used to locally solve free-choice conflicts, is that of assigning a *fixed ratio* of fluid volume to all enabled continuous transitions outputting from an empty continuous place. As an example, in Figure 7 we may assign a ratio $v_{out,1} = s \cdot v_{out,2}$. This new constraint can be added to the set $\mathcal{S}$ or even better, by substitution we can reduce by one the number of variables in (4).

We can also consider the case of *local priority rules* by suitable modification of the linear set (4). Assume that in Figure 7 a legal solution is such that $t_{out,1}$ has priority over $t_{out,2}$, i.e., all fluid entering the place should be consumed by $t_{out,1}$ and only if $v_{out,1} = V_{out,1}$ the remaining fluid should be consumed by $t_{out,2}$. This can be done adding the following constraints:

$$\begin{cases} M \cdot x \geq V_{out,1} - v_{out,1} \\ v_{out,2} \leq M \cdot (1 - x) \end{cases}$$

where $x \in \{0, 1\}$, $M \in \mathbb{R}$ with $M >> 0$. Thus if $v_{out,1} < V_{out,1}$ it follows $v_{out,2} = 0$. The problem with this technique is that a simple LPP is transformed into a more complex mixed integer-linear problem.

6 Conclusions

We have used hybrid stochastic Petri nets as fluid models for flexible manufacturing systems. Assuming that the instantaneous firing speeds of continuous transitions are piecewise constant, we have shown that the set of all possible behaviors of the net during a macro–state can be represented by the convex set defined by a system of linear inequalities. The computation of the instantaneous firing speed — and the associated problem of conflict resolution — can be seen as the net counterpart of a performance optimization with global or local objective functions. Future work will explore the use of linear programming sensitivity analysis for parameter optimization of systems described by hybrid stochastic Petri nets.

References

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

[2] H. Alla, R. David, J. Le Bail, "Hybrid Petri Nets," *ECC'91 European Control Conference* (Grenoble, France), pp. 1472–1477, July 1991.

[3] F. Balduzzi, G. Menga, "A State Variable Model for the Fluid Approximation of Flexible Manufacturing Systems," *Proc. 1998 IEEE Int. Conf. on Robotics and Automation* (Leuven, Belgium), pp. 1172–1178, May 1998.

[4] R.E. Burkard, F. Rendl, "Lexicographic bottleneck problems," *Operations Research Letters*, Vol. 10, pp. 303–308, 1991.

[5] J.A. Buzacott, L.E. Hanifin, "Models of automatic transfer lines with inventory banks: A review and comparison," *AIIE Trans.*, Vol. 10, No. 2, pp. 197–207, 1978.

[6] E. Dubois, H. Alla, R. David, "Continuous Petri Nets with Maximal Speeds Depending on Time," *4th RPI Int. Conf. on Computer Integrated Manufacturing and Automation Technology* (Troy, New York, USA), pp. 32–39, October 1994.

[7] A. Giua, E. Usai "High-Level Hybrid Petri Nets: a Definition," *Proc. 35th Conf. on Decision and Control* (Kobe, Japan), pp. 148–150, December, 1996.

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

[9] A. Mandelbaum, H. Chen, "Discrete Flow Networks: Bottleneck Analysis and Fluid Approximations," *Math. Opns. Res.*, Vol. 16, pp. 408–446, 1991.

[10] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Proceedings IEEE*, Vol. 77, No. 4, pp. 541–580, 1989.