A Linear State Variable Model for First–Order Hybrid Petri Nets *

Fabio Balduzzi (†), Giuseppe Menga (†), Alessandro Giua (‡), Carla Seatzu (‡)

(†) Dip. di Automatica ed Informatica, Politecnico di Torino, Italy
email: {balduzzi,menga}@polito.it

(‡) Dip. di Ingegneria Elettrica ed Elettronica, Università di Cagliari, Italy
email: {giua,seatzu}@diee.unica.it

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. This paper provides a framework to describe the overall hybrid net behavior 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.

1 Introduction

First–Order Hybrid Petri Nets (FOHPN) are nets that consist of continuous places holding fluid, discrete places containing a non–negative integer number of tokens, and transitions, either discrete or continuous. This hybrid Petri net model has been introduced by the authors in [3, 4] and follows the formalism described by David and Alla [2, 6].

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 first–order fluid models, i.e., models in which the continuous flows have constant rates and the fluid content of each continuous place varies linearly with time. Each model is relative to a given *macro–state* that defines the set of admissible *instantaneous firing speed* (IFS) vectors of continuous transitions. Any admissible IFS vector represents a possible mode of operation of the net. Since we consider the first–order behavior, we assume that the IFS vector remains constant within a macro–state. The set of all admissible IFS vectors is characterized by the feasible solutions of a linear constraint set $\mathcal{S}$ as discussed in details in [3, 4].

A discrete–event model describes the behavior of the net that, upon the occurrence of *macro–*

*Published as: F. Balduzzi, G. Menga, A. Giua, C. Seatzu, "A Linear State Variable Model for First-Order Hybrid Petri Nets," *Proc. 14th IFAC World Congress* (Beijing, China), Vol. J, pp. 205-210, July, 1999.

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 [3] the authors have considered two types of macro–events: (a) the firing of a discrete transition; (b) the emptying of a continuous place. In fact, the occurrence of any such event modifies the set $\mathcal{S}$. However, the timing structure associated to the macro–event occurrence has not been explicitly examined in [3].

In this paper we extends previous results in several ways.

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. In this paper we adopt a *single–server enabling–memory* timer policy. This policy is quite common when dealing with timed Petri nets [1]. However other policies may also be associated to FOHPN with minor changes.

We provide an algebraic framework to describe the overall net behavior that combines both time–driven and event–driven dynamics. The resulting overall model is a linear discrete–time time–varying state variable model whose sampling instants are given by the occurrence of the macro–events.

The state vector $\mathbf{x}(k)$ 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 $\mathbf{u}(k)$ is given by the length of the current macro–period and by the characteristic vector that specifies which transition (if any) fires at the end of the current macro–period. First we derive an open–loop formulation of this model and then, by explicitly writing the relation between the input vector and the state, we derive a closed–loop model, in which a reference input is used. Finally we provide the core of a simulation tool, i.e., the algorithm to determine which macro–event will occur next from the given current state.

We see two main advantages in the proposed formulation. Firstly, the linear state variable model can be directly implemented to construct an efficient hybrid simulation tool. Secondly, this algebraic formalism allows one to describe hybrid systems with a well–understood linear (albeit time–varying) state variable model to which classical control theory may be applied (see [5] for a similar approach in the manufacturing domain).

The rest of the paper is structured as follows. In Section 2 we recall the definition of First–Order Hybrid Petri Nets. Section 3 describes the continuous and discrete event dynamics of the net. Section 4 shows how a linear discrete–time time–varying state variable model can be derived. Section 5 presents an algorithm to compute, from a given state, which macro–event will occur next.

2 Background

We recall the Petri net formalism used in this paper following [3, 4]. For a more comprehensive introduction to place/transition Petri nets see [7]. The common notation and semantics for timed nets can be found in [1].

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.

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 the cardinality of the set of timed transition $T_t = T_D \cup T_E$ as q_t. We assume that the transition labelling is such that: $T_t = \{t_j \mid j = 1, \ldots, q_t\}$, $T_I = \{t_j \mid j = q_t + 1, \ldots, q_d\}$, $T_c = \{t_j \mid j = q_d + 1, \ldots, q_d + q_c\}$.

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

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

We require (*well-formed nets*) that for all $t \in T_c$ and for all $p \in P_d$, $Pre(p,t) = Post(p,t)$.

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

The function $\mathcal{C} : T_c \to \mathbb{R}_0^+ \times \mathbb{R}_\infty^+$ specifies the firing speeds associated to continuous transitions (here $\mathbb{R}_\infty^+ = \mathbb{R}^+ \cup \{\infty\}$). For any continuous transition $t_j \in T_c$ we let $\mathcal{C}(t_j) = (V'_j, V_j)$, with $V'_j \leq V_j$. Here V'_j represents the *minimum firing speed* (mfs) and V_j represents the *maximum firing speed* (MFS). In the following, unless explicitly specified, the mfs of a continuous transition 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 $\mathbf{C}(p,t) = Post(p,t) - Pre(p,t)$. 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*

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

is a function that assigns to each discrete place a non-negative 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 a marking at time τ is denoted $\mathbf{m}(\tau)$. The restriction of $\mathbf{m}$ to P_d and P_c are denoted with $\mathbf{m}^d$ and $\mathbf{m}^c$, respectively. An *FOHPN system* $\langle N, \mathbf{m}(\tau_0) \rangle$ is an FOHPN N with an initial marking $\mathbf{m}(\tau_0)$.

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

Definition 1. *Let $\langle N, \mathbf{m} \rangle$ be an FOHPN system. A discrete transition t is* enabled *at $\mathbf{m}$ 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 2. *Let* $\langle N, \mathbf{m}\rangle$ *be an FOHPN system. A continuous transition* t *is* enabled *at* $\mathbf{m}$ *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* $\mathbf{m}$ *if for all places* $p_i \in {}^{(c)}t$, $m_i > 0$*;*
- weakly enabled *at* $\mathbf{m}$ *if for some* $p_i \in {}^{(c)}t$, $m_i = 0$. ■

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

$$\mathbf{e}^T_{i,n} = [\, \underbrace{\overbrace{0, \ldots, 0, 1}^{i}, 0, \ldots, 0}_{n} \,]$$

3 Net dynamics

We describe the hybrid dynamics of an FOHPN considering first the time–driven behavior associated to the firing of continuous transitions, and then the event–driven behavior associated to the firing of discrete transitions.

3.1 Continuous 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

$$\dot{m}_i(\tau) = \sum_{t_j \in T_c} C(p_i, t_j) v_j(\tau) = \mathbf{e}^T_{i,n_c} \mathbf{C}_{cc} \mathbf{v}(\tau) \tag{1}$$

where $\mathbf{v}(\tau) = [v_1(\tau), \ldots, v_{n_c}(\tau)]^T$ is the IFS vector at time τ. Indeed Equation 1 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.

We now characterize the set of all admissible IFS vectors.

Definition 3. (admissible IFS vectors)

Let $\langle N, \mathbf{m}\rangle$ *be an FOHPN system. Let* $T_{\mathcal{E}}(\mathbf{m}) \subset T_c$ *(*$T_{\mathcal{N}}(\mathbf{m}) \subset T_c$*) be the subset of continuous transitions enabled (not enabled) at* $\mathbf{m}$*, and* $P_{\mathcal{E}} = \{p_i \in P_c \mid m_i = 0\}$ *be the subset of empty* *continuous places. Any* admissible IFS vector $\mathbf{v}$ *at* $\mathbf{m}$ *is a feasible solution of the following linear set:*

$$\begin{cases} (a) & V_j - v_j \geq 0 & \forall t_j \in T_{\mathcal{E}}(\mathbf{m}) \\ (b) & v_j - V'_j \geq 0 & \forall t_j \in T_{\mathcal{E}}(\mathbf{m}) \\ (c) & v_j = 0 & \forall t_j \in T_{\mathcal{N}}(\mathbf{m}) \\ (d) & \sum_{t_j \in T_{\mathcal{E}}} C(p, t_j) v_j \geq 0 & \forall p \in P_{\mathcal{E}}(\mathbf{m}) \end{cases} \tag{2}$$

The set of all feasible solutions is denoted $\mathcal{S}(N, \mathbf{m})$*.* ■

Constraints of the form (2.a), (2.b), and (2.c) follow from the firing rules of continuous transitions. Constraints of the form (2.d) follow from (1), because if a continuous place is empty then its fluid content cannot decrease.

Note that the set $\mathcal{S}$ is a function of the marking of the net. Thus as $\mathbf{m}$ changes it may vary as well. In particular it changes at the occurrence of the following macro–events: (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.

Let τ_k and τ_{k+1} be the occurrence times of two consecutive macro–events of this kind; we assume that within the interval of time $[\tau_k, \tau_{k+1})$ the IFS vector is constant and we denote it $\mathbf{v}(\tau_k)$. Then the continuous behavior of an FOHPN for $\tau \in [\tau_k, \tau_{k+1})$ is described by

$$\begin{cases} \mathbf{m}^c(\tau) & = \mathbf{m}^c(\tau_k) + \mathbf{C}_{cc}\mathbf{v}(\tau_k)(\tau - \tau_k) \\ \mathbf{m}^d(\tau) & = \mathbf{m}^d(\tau_k) \end{cases} \tag{3}$$

3.2 Discrete event dynamics

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

Definition 4. *Let* $\langle N, \mathbf{m} \rangle$ *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*

$$\nu_j(\tau) = \nu_j(\tau_k) + (\tau - \tau_k), \tag{4}$$

while if t_j *is not enabled in this interval then*

$$\nu_j(\tau) = \nu_j(\tau_k) = 0. \tag{5}$$

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

With the notation of [1], 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 $\boldsymbol{\nu} \in \mathbb{R}^{q_t}$. Note that the timer dynamics is piecewise constant 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 $\nu_j(\tau) = \hat{\nu}_j$. 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. An immediate transition fires as soon as it is enabled, i.e., it can be considered as a deterministic transition with $\hat{\nu} = 0$.

The firing of a discrete transition t_j at $\mathbf{m}(\tau)$ yields the marking

$$\begin{cases} \mathbf{m}^c(\tau) & = \mathbf{m}^c(\tau^-) + \mathbf{C}_{cd}\boldsymbol{\sigma}(\tau) \\ \mathbf{m}^d(\tau) & = \mathbf{m}^d(\tau^-) + \mathbf{C}_{dd}\boldsymbol{\sigma}(\tau) \end{cases} \tag{6}$$

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

4 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.

4.1 Macro–events, macro–period 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 or disable a set of continuous (discrete) transitions, thus modifying the set $\mathcal{S}$ (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 [3, 4] 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 $\mathbf{m}$, the timer vector $\boldsymbol{\nu}$, the constraint set $\mathcal{S}$ are summarized in the table below.

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

Let τ_k 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 $\Delta(k) = (\tau_{k+1} - \tau_k)$. 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 [1].

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

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{m}^c(\tau_k) \\ \mathbf{m}^d(\tau_k) \\ \boldsymbol{\nu}(\tau_k) \end{bmatrix} \begin{matrix} \} \, n_c \\ \} \, n_d \\ \} \, q_t \end{matrix} \tag{7}$$

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

4.2 Discrete–time open–loop dynamics

We now derive a discrete–time state evolution law for the state vector (7) that can be expressed in matrix notation as:

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k), \tag{8}$$

where $\mathbf{u}(k) \in \mathbb{R}^{q_d+1}$ is the input vector, $\mathbf{A}(k)$ and $\mathbf{B}(k)$ 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:

$$\begin{cases} \mathbf{m}^c(k+1) &= \mathbf{m}^c(k) + \mathbf{C}_{cc}\mathbf{v}(k)\Delta(k) \\ & \quad +\mathbf{C}_{cd}\boldsymbol{\sigma}(k+1) \\ \mathbf{m}^d(k+1) &= \mathbf{m}^d(k) + \mathbf{C}_{dd}\boldsymbol{\sigma}(k+1) \\ \boldsymbol{\nu}(k+1) &= \mathbf{D}(k)\boldsymbol{\nu}(k) + \mathbf{f}(k)\Delta(k). \end{cases} \tag{9}$$

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 the event occurring at time τ_{k+1} and $\Delta(k)$ is the length of the macro–period.

The first two equations follow from the combination of the net dynamic equations (3), (6).

The third equation follows from (4) and (5). Here matrix $\mathbf{D}(k) \in \mathbb{R}^{q_t \times q_t}$ and vector $\mathbf{f}(k) \in \mathbb{R}^{q_t}$ 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 $t_{\bar{j}} \in T_t$

$$f_{\bar{j}}(k) = \begin{cases} 1 & \text{if } t_{\bar{j}} \text{ is enabled at } \mathbf{m}(k) \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{D}(k) = \text{diag}\{\mathbf{f}(k)\} \tag{10}$$

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

- *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 $t_{\bar{\jmath}} \in \overline{T}_j$, $\nu_{\bar{\jmath}}(k+1)$ will be reset to 0 regardless of the value of $\nu_{\bar{\jmath}}(k)$. Furthermore the timer of t_j will also be reset to 0 after its firing. Therefore for all $t_{\bar{\jmath}} \in T_t$

$$f_{\bar{\jmath}}(k) = \begin{cases} 1 & \text{if } (j \neq \bar{\jmath}) \ \wedge \ (t_{\bar{\jmath}} \notin \overline{T}_j) \\ & \wedge \ t_{\bar{\jmath}} \text{ is enabled at } \mathbf{m}(k) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$\mathbf{D}(k) = \mathrm{diag}\,\{\mathbf{f}(k)\}$$

- *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 $t_{\bar{\jmath}} \in T_t$

$$f_{\bar{\jmath}}(k) = \begin{cases} 1 & \text{if } (t_{\bar{\jmath}} \notin \overline{T}_j) \\ & \wedge \ t_{\bar{\jmath}} \text{ is enabled at } \mathbf{m}(k) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$\mathbf{D}(k) = \mathrm{diag}\,\{\mathbf{f}(k)\}$$

Finally, we observe that equation (9) is in the form of equation (8) if we consider (7) and let

$$\mathbf{A}(k) = \begin{bmatrix} \mathbf{I}_{n_c \times n_c} & \mathbf{0}_{n_c \times n_d} & \mathbf{0}_{n_c \times q_t} \\ \mathbf{0}_{n_d \times n_c} & \mathbf{I}_{n_d \times n_d} & \mathbf{0}_{n_d \times q_t} \\ \mathbf{0}_{q_t \times n_c} & \mathbf{0}_{q_t \times n_d} & \mathbf{D}(k) \end{bmatrix}, \tag{13}$$

$$\mathbf{B}(k) = \begin{bmatrix} \mathbf{C}_{cc}\mathbf{v}(k) & \mathbf{C}_{cd} \\ \mathbf{0}_{n_d} & \mathbf{C}_{dd} \\ \mathbf{f}(k) & \mathbf{0}_{q_t \times n_d} \end{bmatrix}, \tag{14}$$

and

$$\mathbf{u}(k) = \begin{bmatrix} \Delta(k) \\ \boldsymbol{\sigma}(k+1) \end{bmatrix}. \tag{15}$$

The input vector $\mathbf{u}$ specifies: (a) the length $\Delta(k)$ of the current macro–period; (b) which transition (if any) will fire at the end of the current macro–period. Note that $\Delta(k)$ and $\boldsymbol{\sigma}(k+1)$ depend on the state vector $\mathbf{x}(k)$ 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,

$$\Delta(k) = \frac{-m_i(k)}{\dot{m}_i(k)} = \frac{-\mathbf{e}_{i,s}^T \mathbf{x}(k)}{\mathbf{e}_{i,n_c}^T \mathbf{C}_{cc}\mathbf{v}(k)}; \tag{16}$$

$$\boldsymbol{\sigma}(k+1) = \mathbf{0}_{q_d}. \tag{17}$$

- *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.,

$$\Delta(k) = \hat{\nu}_j - \nu_j(k) = \hat{\nu}_j - \mathbf{e}_{n+j,s}\mathbf{x}(k). \tag{18}$$

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

$$\boldsymbol{\sigma}(k+1) = \mathbf{e}_{j,q_d}. \tag{19}$$

- *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 $C(p_i, t_j)$ 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,

$$\begin{aligned}\Delta(k) &= \frac{C(p_i,t_j) - m_i(k)}{\dot{m}_i(k)} \\ &= \frac{C(p_i,t_j) - \mathbf{e}_{i,s}^T\mathbf{x}(k)}{\mathbf{e}_{i,n_c}^T\mathbf{C}_{cc}\mathbf{v}(k)};\end{aligned} \tag{20}$$

$$\boldsymbol{\sigma}(k+1) = \mathbf{0}_{q_d}. \tag{21}$$

4.3 Discrete–time closed–loop dynamics

In this subsection we derive a closed–loop form of equation (8) by substituting the values of $\Delta(k)$ and $\boldsymbol{\sigma}(k+1)$ in equation (15).

We can write

$$\mathbf{u}(k) = \mathbf{r}(k) - \mathbf{K}(k)\mathbf{x}(k)$$

where $\mathbf{r}(k) \in \mathbb{R}^{q_d+1}$ is the set point vector and $\mathbf{K}(k) \in \mathbb{R}^{s\times(q_d+1)}$ is the feedback gain matrix.

The set point vector $\mathbf{r}(k)$ depends on the macro-event occurring at the sampling instant $k+1$ and is defined as follows:

- *Macro-event* π_i. All components of $\mathbf{r}(k)$ are zero.

$$\mathbf{r}(k) = \begin{bmatrix} 0 \\ \mathbf{0}_{q_d} \end{bmatrix}. \tag{22}$$

- *Macro-event* γ_j. The first component of $\mathbf{r}(k)$ is the firing delay $\hat{\nu}_j$ of transition t_j — we assume $\hat{\nu}_j = 0$ is t_j is an immediate transition — while $\boldsymbol{\sigma}(k+1) = \mathbf{e}_{j,q_d}$ is the firing count vector associated to the firing of t_j.

$$\mathbf{r}(k) = \begin{bmatrix} \hat{\nu}_j \\ \mathbf{e}_{j,q_d} \end{bmatrix}. \tag{23}$$

- *Macro-events* ε_i, $\bar{\varepsilon}_i$. For macro event ε_i the first component of $\mathbf{r}(k)$ is the time the continuous place p_i takes to reach the fluid level $C(p_i, t_j)$ that causes the event occurrence if its initial marking were zero and the IFS vector were $\mathbf{v}(k)$. For macro event $\bar{\varepsilon}_i$ the first component of $\mathbf{r}(k)$ is the time changed of sign the continuous place p_i takes to reach the fluid level zero if its initial marking had the value $C(p_i, t_j)$ that causes the event occurrence and the IFS vector were $\mathbf{v}(k)$. In both cases, vector $\mathbf{0}_{q_d}$ indicates that no discrete transitions fires.

$$\mathbf{r}(k) = \begin{bmatrix} \dfrac{C(p_i, t_j)}{\mathbf{e}_{i,n_c}^T \mathbf{C}_{cc}\mathbf{v}(k)} \\ \mathbf{0}_{q_d} \end{bmatrix}. \tag{24}$$

If we let $\bar{\mathbf{A}}(k) = \mathbf{A}(k) - \mathbf{K}(k)\mathbf{B}(k)$ we obtain the following closed–loop equation

$$\mathbf{x}(k+1) = \bar{\mathbf{A}}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{r}(k). \tag{25}$$

We observe that $\mathbf{B}(k)$ is the same as in the open–loop equation and its expression is given by (14).

Matrix $\bar{\mathbf{A}}(k)$ depends on the macro-event occurring at the sampling instant $k+1$ and is defined as follows:

- *Macro-events* π_i, ε_i, $\bar{\varepsilon}_i$.

$$\bar{\mathbf{A}}(k) = \mathbf{A}(k) + \begin{bmatrix} \dfrac{-\mathbf{C}_{cc}\mathbf{v}(k)\mathbf{e}_{i,s}^T}{\mathbf{e}_{i,n_c}^T \mathbf{C}_{cc}\mathbf{v}(k)} \\ \mathbf{0}_{n_d \times s} \\ \dfrac{-\mathbf{f}(k)\mathbf{e}_{i,s}^T}{\mathbf{e}_{i,n_c}^T \mathbf{C}_{cc}\mathbf{v}(k)} \end{bmatrix}. \tag{26}$$

- *Macro-event* γ_j:

$$\bar{\mathbf{A}}(k) = \mathbf{A}(k) + \begin{bmatrix} -\mathbf{C}_{cc}\mathbf{v}(k)\mathbf{e}_{n+j,s}^T \\ \mathbf{0}_{n_d \times s} \\ -\mathbf{f}(k)\mathbf{e}_{n+j,s}^T \end{bmatrix}. \tag{27}$$

5 Macro-behaviour: a simulation algorithm

In the previous section a linear time-varying, discrete-time state variable model has been derived to describe the macro-behaviour of an FOHPN. However, the above model has been obtained under the assumption that at each discrete sampling instant k, the occurrence of the next macro-event is known. For this purpose, we provide a simulation algorithm to determine, given the actual state $\mathbf{x}(k)$, which is the next macro–event to occur.

1. Let $\Psi_k = \emptyset$. This set will contain all pairs (α, Δ_α), where α is an event that may potentially occur and Δ_α is its residual lifetime.

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

3. If $\Psi_k \neq \emptyset$, then goto 8.

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

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

6. For each discrete transition t_j that is not enabled at $\mathbf{x}(k)$, let

$$P_j = \{p_\ell \in {}^{(c)}t_j \mid m_\ell(k) < C(p_\ell, t_j)\};$$

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.

   (a) $\mathbf{m}^d(k) \geq \mathbf{C}_{dd}(\cdot, t_j)$, i.e., t_j is enabled in the discrete sub–net;

   (b) $\forall p_\ell \in P_j$, $\dot{m}_\ell(k) = \mathbf{e}_{\ell,n_c}^T \mathbf{C}_{cc}\mathbf{v}(k) > 0$, i.e., the marking of all places in P_j is increasing.

   The time it takes for t_j to become enabled is

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

   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

$$\bar{P}_j = \{p_\ell \in {}^{(c)}t_j \mid \begin{array}{l} m_\ell(k) \geq C(p_\ell, t_j), \\ \dot{m}_\ell(k) < 0\}, \end{array}$$

   will go below the fluid level $C(p_{\bar{\ell}}, t_j)$ in the meantime. Thus we let

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

   with $\bar{\Delta} = \infty$ if $\bar{P}_j = \emptyset$.

   If $\bar{\Delta} > \Delta$, add (ε_i, Δ) to Ψ_k.

7. 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

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

   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 $C(p_i, t_j)$. Thus we add $(\bar{\varepsilon}_i, \Delta^+)$ to Ψ_k.

8. Choose from Ψ_k the pair (α, Δ_α) where Δ_α is the minimum over all pairs. Event α is the next to occur.

   Note that if two pairs (α_1, Δ) and (α_2, Δ^+) are in Ψ_k, then $\Delta < \Delta^+)$ and α_1 should be chosen.

9. Let

$$\mathbf{x}(k+1) = \bar{\mathbf{A}}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{r}(k)$$

   where matrices $\bar{\mathbf{A}}(k)$, $\mathbf{B}(k)$ and vector $\mathbf{r}(k)$ are defined in accordance with the previous results and depend on the type of macro-event α.

6 Conclusions

We have considered in this paper First–Order Hybrid Petri Nets, and we have shown how it is possible to describe the overall hybrid net behavior that combines both time–driven and event–driven dynamics with a linear discrete–time time-varying state variable model. This model can be directly used by an efficient simulation tool. Furthermore, using this formulation, classical control theory results may potentially be applied to study properties of hybrid systems and this will be the subject of future work.

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," *European Control Conference* (Grenoble, France), pp. 1472–1477, Jul 1991.

[3] F. Balduzzi, A. Giua, G. Menga, "Hybrid Stochastic Petri Nets: Firing Speed Computation and FMS Modelling," *WODES'98, Proc. Fourth Workshop on Discrete Event Systems*, (Cagliari, Italy), pp. 432–438, Aug 1998.

[4] F. Balduzzi, A. Giua, G. Menga, "Optimal Speed Allocation and Sensitivity Analysis of Hybrid Stochastic Petri Nets," *Proc. 1998 IEEE Conf. on Systems, Man and Cybernetics*, (San Diego, California), pp. 656–662, Oct 1998.

[5] F. Balduzzi, A. Giua, G. Menga, "Hybrid Analisys of Automated Manufacturing Systems Using Discrete Linear Inclusions," *Proc. 37th IEEE Conf. Decision and Control*, (Tampa, Florida), pp. 1710–1715, Dec 1998.

[6] R. David, H. Alla, "Autonomous and Timed Continuous Petri Nets," *11th Int. Conf. on Application and Theory of Petri Nets* (Paris, France), pp. 367–386, Jun 1990.

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