HYPENS Manual

Fausto Sessego, Alessandro Giua, Carla Seatzu*

February 7, 2008

HYPENS is an open source tool to simulate timed discrete, continuous and hybrid Petri nets. It has been developed in Matlab to allow designer and user to take advantage of several functions and structures already defined in Matlab, such as optimization routines, stochastic functions, matrices and arrays, etc. The tool can also be easily interfaced with other Matlab programs and be used for analysis and optimization via simulation. The large set of plot functions available in Matlab allow one to represent the results of the simulation in a clear and intuitive way.

1 Background on Hybrid Petri nets

We recall the FOHPN formalism used by HYPENS, following [3].

Net structure: A FOHPN is a structure \( N = (P, T, \text{Pre}, \text{Post}, \mathcal{D}, \mathcal{C}) \).

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 cardinality of \( T \), \( T_d \) and \( T_c \) is denoted \( q \), \( q_d \) and \( q_c \).

The pre- and post-incidence functions that specify the arcs are (here \( \mathbb{R}_0^+ = \mathbb{R}^+ \cup \{0\} \)): \( \text{Pre}, \text{Post} : P_c \times T \rightarrow \mathbb{R}_0^+, P_d \times T \rightarrow \mathbb{N} \). We require that \( \forall t \in T_c \) and \( \forall p \in P_d \), \( \text{Pre}(p, t) = \text{Post}(p, t) \), so that the firing of continuous transitions does not change the marking of discrete places.

Transitions in \( T_d \) may either be deterministic or stochastic. In the case of deterministic transitions the function \( \mathcal{D} : T_d \rightarrow \mathbb{R}_0^+ \) specifies the timing associated to timed discrete transitions. In the case of stochastic transitions \( \mathcal{D} \) defines the parameter(s) of the distribution function corresponding to the timing delay. 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 \( V'_j \leq V_j \): \( V'_j \) represents the minimum firing speed (mfs), \( V_j \)

*F. Sessego, A. Giua and C. Seatzu are with the Department of Electrical and Electronic Engineering, University of Cagliari, \{fausto.sessego,giua,seatzu\}@diee.unica.it.
represents the maximum firing speed (MFS).

The incidence matrix of the net is defined as $C(p, t) = Post(p, t) - Pre(p, t)$. The restriction of $C$ ($Pre$, $Post$, resp.) to $P_x$ and $T_y$, with $x, y \in \{c,d\}$, is denoted $C_{xy}$ ($Pre_{xy}$, $Post_{xy}$, resp.).

A marking is a function that assigns to each discrete place a non-negative number of tokens, and to each continuous place a fluid volume. Therefore $M : P_c \rightarrow \mathbb{R}_0^+$, $P_d \rightarrow \mathbb{N}$. The marking of place $p_i$ is denoted $M_i$, while the value of the marking at time $\tau$ is denoted $M(\tau)$. The restriction of $M$ to $P_d$ and $P_c$ are denoted with $M^d$ and $M^c$, resp.

A system $\langle N, M(\tau_0) \rangle$ is an FOHPN $N$ with an initial marking $M(\tau_0)$.

**Net dynamics:** The enabling of a discrete transition depends on the marking of all its input places, both discrete and continuous. More precisely, a discrete transition $t$ is enabled at $M$ if for all $p_i \in t^+$, $M_i \geq Pre(p_i, t)$, where $t^+$ denotes the preset of transition $t$. The enabling degree of $t$ at $M$ is equal to $enab(M, t) = \max\{k \in \mathbb{N} \mid M \geq k \cdot Pre(\cdot)\}$.

If $t$ is infinite-server semantics, we associate to it a number of clocks that is equal to $enab(M, t)$. Each clock is initialized to a value that is equal to the time delay of $t$, if $t$ is deterministic, or to a random value depending on the distribution function of $t$, if $t$ is stochastic. If a discrete transition is $k$-server semantics, then the number of clocks that are associated to $t$ is equal to $\min\{k, enab(M, t)\}$. The values of clocks associated to $t$ decrease linearly with time, and $t$ fires when the value of one of its clocks is null (if $k$ clocks reach simultaneously a null value, then $t$ fires $k$ times). Note that here we are considering enabling memory policy, not total memory policy. This means that if a transition enabling degree is reduced by the firing of a different transition, then the disabled clocks have no memory of this in future enabling [1, 4].

If a discrete transition $t_j$ fires $k$ times at time $\tau$, then its firing at $M(\tau^-)$ yields a new marking $M(\tau)$ such that $M^c(\tau) = M^c(\tau^-) + C_{cd}\sigma$, and $M^d(\tau) = M^d(\tau^-) + C_{dd}\sigma$, where $\sigma = k \cdot e_j$ is the firing count vector associated to the firing of transition $t_j$ $k$ times.

To every continuous transition $t_j$ is associated an instantaneous firing speed (IFS) $v_j(\tau)$. For all $\tau$ it should be $V_j^-' \leq v_j(\tau) \leq V_j$, and the IFS of each continuous transition is piecewise constant between events.

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 strong and weakly enabled: if all input continuous places of $t_j$ have a not null marking, then $t_j$ is called strongly enabled, else $t_j$ is called weakly enabled.

We can write the equation which governs the evolution in time of the marking of a place $p_i \in P_c$ as $\dot{m}_i(\tau) = \sum_{t_j \in T_c} C(p_i, t_j)v_j(\tau)$ where $v(\tau) = [v_1(\tau), \ldots, v_{n_c}(\tau)]^T$ is the IFS vector at time $\tau$.

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 $t_j$

---

1We are using an enabling policy for continuous transitions slightly different from the one proposed by David and Alla [4]. See [2] for a detailed discussion.
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. Linear inequalities can be used to characterize the set of all admissible firing speed vectors \( S \). Each vector \( v \in S \) 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, i.e., the one that maximize a given performance index \( J(v) \) [2].

We say that a macro-event (ME) 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; (c) a continuous place, whose marking is increasing (decreasing), reaches a flow level that increases (decreases) the enabling degree of discrete transitions.

Let \( \tau_k \) and \( \tau_{k+1} \) be the occurrence times of two consecutive ME as defined above; we assume that within the interval of time \([\tau_k, \tau_{k+1}]\), denoted as a macro-period (MP), the IFS vector is constant and we denote it \( v(\tau_k) \). Then the continuous behavior of an FOHPN for \( \tau \in [\tau_k, \tau_{k+1}] \) is described by \( M^c(\tau) = M^c(\tau_k) + C_{cc} v(\tau_k)(\tau - \tau_k) \), \( M^d(\tau) = M^d(\tau_k) \).

Example 1.1 Consider the net system in Fig. 1.a. Place \( p_1 \) is a continuous place, while all other places are discrete. Continuous transitions \( t_1, t_2 \) have MFS \( V_1 = 1, V_2 = 2 \) and null mfs. Deterministic timed discrete transitions \( t_3, t_5 \) have timing delays 2 and 1.5, resp. Exponential stochastic discrete transitions \( t_4, t_6 \) have average firing rates are \( \lambda_4 = 2 \) and \( \lambda_6 = 1.5 \).

The continuous transitions represent two unreliable machines; parts produced by the first machine (\( t_1 \)) are put in a buffer (\( p_1 \)) before being processed by the second machine (\( t_2 \)). The discrete subnet represents the failure model of the machines. When \( p_3 \) is marked, \( t_1 \) is enabled, i.e. the first machine is operational; when \( p_2 \) is marked, transition \( t_1 \) is not enabled, i.e. the first machine is down. A similar interpretation applies to the second machine.

Assume that we want to maximize the production rates of machines. In such a case, during any MP continuous transitions fire at their highest speed. This means that we want to maximize \( J(v) = v_1 + v_2 \) under the constraints \( v_1 \leq V_1, v_2 \leq V_2 \), and — when \( p_1 \) is empty — \( v_2 \leq v_1 \).

The resulting evolution graph and the time evolution of \( M_1 \), \( v_1 \) and \( v_2 \) are shown in Fig. 1.b and c. During the first MP (of length 1) both continuous transitions are strongly enabled and fire at their MFS. After one time unit, \( p_1 \) gets empty, thus \( t_2 \) becomes weakly enabled fires at the same speed of \( t_1 \). At time \( \tau = 1.5 \), transition \( t_5 \) fires, disabling \( t_2 \).

2 HYPENS Tool

HYPENS has been developed in Matlab (Version 7.1). It is composed of 4 main files. The first two files, make_HPN.m and enter_HPN.m create the net to be simulated: the former requires input data from the workspace while the latter is a guided procedure.

The file simulator_HPN.m computes the timing evolution of the net that is summarized in an
array of cells called `Evol`. Based on this array, the file `analysis_HPN.m` computes useful statistics and plots the simulation results.

**Function make_HPN:**

```
[Pre, Post, M0, vel, v, D, s, alpha] = make_HPN (Precc, Predc, Predd, Postcc, Postcd, Postdd, M0c, M0d, vel, v, D, s, alpha).
```

This function has the following input arguments.

- **Matrices** `Precc`, `Predc`, `Predd`, `Postcc`, `Postcd`, `Postdd`;
- The **initial marking** `M0c` and `M0d` of continuous/discrete places.
- **Matrix** `vel` ∈ $(\mathbb{R}^+_0)^{q_c \times 2}$ specifies, for each continuous transition, the mfs and the MFS.
- **Vector** `v` ∈ $\mathbb{N}^{1 \times q_d}$ specifies the timing structure of each discrete transition. The entries of this vector may take the following values: 1 - deterministic; 2 - exponential distribution; 3 - uniform distribution; 4 - Poisson distribution; 5 - Rayleigh distribution; 6 - Weibull distribution; etc.
- **Matrix** `D` ∈ $(\mathbb{R}^+_0)^{q_d \times 3}$ associates to each discrete transition a row vector of length 3. If the transition is deterministic, the first element of the row is equal to the time delay of transition. If the transition is stochastic, the elements of the row specify the parameters of the corresponding distribution function (up to three, given the available distribution functions).
- **Vector** `s` ∈ $\mathbb{N}^{1 \times q_d}$ keeps track of the number of servers associated to discrete transitions. The entries take any value in $\mathbb{N}$: 0 if the corresponding transition has infinite servers; $k > 0$ if the corresponding transition has $k$ servers.
- **Vector** `alpha` specifies the conflict resolution policy among discrete transitions.
• When $\alpha \in \mathbb{N}^{1 \times q_d}$ then two cases are possible. If all its entries are zero, conflict resolution is solved by priorities that depend on the indices of transitions (the smallest the index, the highest the priority). Otherwise, all its entries are greater than zero and specify the weight of the corresponding transition, i.e., if $T_e$ is the set of enabled transitions, the probability of firing transition $t \in T_e$ is $\pi(t) = \alpha(t)/ (\sum_{t' \in T_e} \alpha(t'))$.

• When $\alpha \in \mathbb{N}^{2 \times q_d}$ the first row specifies the weights associated to transitions (as in the previous case) while the second row specifies the priorities associated to transitions. During simulation, when a conflict arises, priority are first considered; in the case of equal priority, weights are used to solve the conflict. See [4] for details.

Output data of function $\text{enter}_{\text{HPN}}$ are nothing else than input data, appropriately rewritten so as to constitute the input to function $\text{simulator}_{\text{HPN}}$.

- Matrices $\text{Pre}$ and $\text{Post}$ are defined as:

\[
\text{Pre} = \begin{bmatrix}
\text{Pre}_{ec} & \text{NaN} & \text{Pre}_{cd} \\
\text{NaN} & \text{NaN} & \text{NaN} \\
\text{Pre}_{dc} & \text{NaN} & \text{Pre}_{dd}
\end{bmatrix}, \quad \text{Post} = \begin{bmatrix}
\text{Post}_{ec} & \text{NaN} & \text{Post}_{cd} \\
\text{NaN} & \text{NaN} & \text{NaN} \\
\text{Post}_{dc} & \text{NaN} & \text{Post}_{dd}
\end{bmatrix}
\]

where a row and a column of $\text{NaN}$ (not a number) have been introduced so as to better visualize the continuous and/or discrete sub-matrices.

- The initial marking is denoted as $M_0$ and is defined as a column vector.

- All other output data are identical to the input data.

**Function enter\_HPN:** $[\text{Pre}, \text{Post}, M_0, \text{vel}, v, D, s, \alpha] = \text{enter}_{\text{HPN}}$.

This function creates the net following a guided procedure. The parameters are identical to those defined for the previous function $\text{make}_{\text{HPN}}.m$.

**Function simulator\_HPN:** $\text{Evol} = \text{simulator}_{\text{HPN}}(\text{Pre}, \text{Post}, M, \text{vel}, v, D, s, \alpha, \text{time\_stop}, \text{simulation\_type}, J)$.

The input data of function $\text{simulator}_{\text{HPN}}$ coincide with the output data of the previous interface functions, plus three additional parameters.

- $\text{time\_stop}$ is equal to the time length of simulation.

- $\text{simulation\_type} \in \{2, 1, 0\}$ specifies the simulation mode. Mode 0: no intermediate result is shown but only array $\text{Evol}$ is created to be later analyzed by function $\text{analysis}_{\text{HPN}}$. Modes 1 and 2 generate on screen the evolution graph: in the fist case the simulation proceeds without interruptions until $\text{time\_stop}$ is reached; in the second case the simulation is carried out step-by-step.

- $J \in \mathbb{R}^{1 \times q_c}$ is a row vector that associates to each continuous transition a weight: $J \cdot v$ is the linear cost function that should be maximized at each MP to compute the IFS vector $v$. This
optimization problem is solved using the subroutine \texttt{glpkmex.m} of Matlab.

Output data are saved in an array of cells called \textit{Evol}, with the following entries (here \( K \) is the number of ME that occur during the simulation run).

- \( \text{Type} \in \{1, 2, 3\} \): 1 (2, 3) if the net is continuous (discrete, hybrid).

- \( M_{Evol} \in (\mathbb{R}_0^+)^{n \times (K+1)} \) keeps track of the marking of the net during all the evolution. In particular, an \( n \)-dimensional column vector is associated to the initial time instant and to all the time instants in which a different ME occurs, each one representing the corresponding value of the marking at that time instant.

- \( IFS_{Evol} \in (\mathbb{R}_0^+)^{q_c \times (K+1)} \) keeps track of the IFS vectors during all the evolution. In particular, a \( q_c \)-dimensional column vector is associated to the initial configuration and to the end of each ME.

- \( P_{macro} \) and \( Event_{macro_{Evol}} \) are \((K+1)\)-dimensional row vectors and keep track of the ME caused by continuous places. If the generic \( r \)-th ME is due to continuous place \( p_j \), then the \((r+1)\)-th entry of \( P_{macro} \) is equal to \( j \); if it is due to the firing of a discrete transition, then the \((r+1)\)-th entry of \( P_{macro} \) is equal to NaN. The entries of \( Event_{macro_{Evol}} \) may take values in \( \{0, 1, -1, NaN\} \): 0 means that the corresponding continuous place gets empty; 1 means that the the continuous place enables a new discrete transition; \(-1\) means that the continuous place disables a discrete transition. If the generic \( r \)-th ME is due to the firing of a discrete transition, then the \((r+1)\)-th entry of \( Event_{macro_{Evol}} \) is equal to NaN as well. The first entries of both \( P_{macro} \) and \( Event_{macro_{Evol}} \) are always equal to NaN.

- \( firing_{transition} \) is a \((K+1)\)-dimensional row vector that keeps track of the discrete transitions that have fired during all the evolution. If the \( r \)-th ME is caused by the firing of discrete transition \( t_k \), then the \((r+1)\)-th entry of \textit{firing\_transition} is equal to \( k \); if it is caused by a continuous place, then the \((r+1)\)-th entry of \textit{firing\_transition} is equal to 0. Note that the first entry of \textit{firing\_transition} is always equal to NaN.

- \( timer_{macro\_event} \) is a \((K+1)\)-dimensional row vector that keeps into memory the length of ME. The first entry is always equal to NaN.

- \( \tau \) is equal to the total time of simulation.

- \( Q_{Evol} \) is a \(((K+1) \times q_d)\)-dimension array of cells, whose generic \((r+1, j)\)-th entry specifies the clocks of transition \( t_j \) at the end of the \( r \)-th MP.

- \( P_c P_d T_c T_d \in \mathbb{N}^4 \) is a 4-dimensional row vector equal to \([n_c \ n_d \ q_c \ q_d]\).

\textbf{Function analysis\_HPN: } \([P_{ave}, P_{max}, P_{ave\_t}, IFS_{ave}, Td_{ave},
Pd_{freq}, Md_{freq}] = \text{analysis}\_HPN (Evol, static\_plot, graph, marking\_plot, Td\_firing\_plot, up\_marking\_plot,
Pd\_prob\_plot, Pd\_ave\_t\_plot, IFS\_plot,
up\_IFS\_plot, IFS\_ave\_plot, Td\_freq\_plot)\).

This function computes useful statistics and plots the results of the simulation run contained in
Evol. The other input parameters are the following.

- $\text{statistic.plot} \in \{0, 1\}$: if 1 two histograms are created showing for each place, the maximum and the average value of the marking during the simulation.

- $\text{graph} \in \{0, 1\}$: if 1 the evolution graph is printed on screen.

- $\text{marking.plot}$: is a vector used to plot the marking evolution of selected places in separate figures. As an example, if we set $\text{marking.plot} = [x \ y \ z]$, the marking evolution of places $p_x$, $p_y$ and $p_z$ is plotted. If $\text{marking.plot} = [-1]$, then the marking evolution of all places is plotted.

- $\text{Td.firing.plot} \in \{0, 1\}$: if 1 a graph is created showing the time instants at which discrete transitions have fired.

- $\text{up.marking.plot}$ is a vector used to plot the marking evolution of selected places in a single figure. The syntax is the same as that of $\text{marking.plot}$.

- $\text{Pd.prob.plot} \in \{0, 1\}$: 1 means that as many plots as the number of discrete places will be visualized, each one representing the frequency of having a given number of tokens during the simulation run.

- $\text{Pd.ave.t.plot}$ is a vector used to plot the average marking in discrete places with respect to time. A different figure is associated to each place, and the syntax to select places is the same as that of $\text{marking.plot}$.

- $\text{IFS.plot}$ (resp., $\text{up.IFS.plot}$): is a vector used to plot the IFS of selected transitions in separate figures (resp., in a single figure). The syntax is the same as that of $\text{marking.plot}$ and $\text{up.marking.plot}$.

- $\text{IFS.ave.plot} \in \{0, 1\}$: if 1 an histogram is created showing the average firing speed of continuous transitions.

- $\text{Td.freq.plot} \in \{0, 1\}$: if 1 an histogram is created showing the firing frequency of discrete transitions during the simulation run.

These are the outputs of function $\text{analysis.HPN.m}$.

- $\text{P.ave}(P_{\text{max}}) \in \mathbb{R}^{1 \times n}$: each entry is equal to the average (maximum) marking of the corresponding place during the simulation run.

- $\text{Pd.ave.t} \in \mathbb{R}^{n_d \times K}$: column $k$ specifies the average marking of discrete places from time $\tau_0 = 0$ to time $\tau_k$ when the $k$-th ME occurs.

- $\text{IFS.ave} \in \mathbb{R}^{1 \times q_c}$: each entry is equal to the average IFS of the corresponding continuous transition.

- $\text{Td.ave} \in \mathbb{R}^{1 \times q_d}$: each entry is equal to the average enabling time of the corresponding discrete transition.

- $\text{Pd.freq} \in \mathbb{R}^{n_d \times (z+1)}$ specifies for each discrete place the frequency of a given number of tokens
during the simulation run. Here \( z \) is the maximum number of tokens in any discrete place during the simulation run.

\(-Md_{\text{freq}} \in \mathbb{R}^{(n_d+1) \times \tilde{K}}\) where \( \tilde{K} \) denotes the number of different discrete markings that are reached during the simulation run. Each column of \( Md_{\text{freq}} \) contains one of such markings and its corresponding frequency as a last entry.

3 A numerical example

Let us consider again the FOHPN system in Fig. 1. For simplicity, we assume that input data using function \textit{make\_HPN.m} are entered in a file \textit{net.m} where we define the following matrices:

\[
\begin{align*}
Pre_{CC} &= \begin{bmatrix} 0 & 1 \end{bmatrix}; & Pre_{CD} &= \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}; \\
Pre_{DC} &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}; & Pre_{DD} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \\
Post_{CC} &= \begin{bmatrix} 1 & 0 \end{bmatrix}; & Post_{CD} &= \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}; \\
Post_{DC} &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}; & Post_{DD} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \\
M_{0C} &= \begin{bmatrix} 0 \end{bmatrix}; & M_{0D} &= \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}; \\
vel &= \begin{bmatrix} 0 & 1 & 0 & 2 \end{bmatrix}; & v &= \begin{bmatrix} 1 & 2 & 1 & 2 \end{bmatrix}; \\
D &= \begin{bmatrix} 2 & NaN & NaN \\ 1.5 & NaN & NaN \\ 1.5 & NaN & NaN \end{bmatrix}; & s &= \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}; \\
alpha &= \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix};
\end{align*}
\]

We save the above input data in an array of cells

\[
\text{net}\_\text{array} = \{ \text{Pre\_CC, Pre\_CD, Pre\_DC, Pre\_DD, Post\_CC, Post\_CD,} \\
\text{Post\_DC, Post\_DD, M\_0C, M\_0D, vel, v, D, s, alpha} \}
\]

Now, input should be rewritten in an appropriate form so as to provide the input data of function \textit{simulator\_HPN.m}. Two different syntax are possible:

1.

\[
\text{correct\_net} = \text{make\_HPN} \left( \text{Pre\_CC, Pre\_CD, Pre\_DC, Pre\_DD, Post\_CC, Post\_CD,} \\
\text{Post\_DC, Post\_DD, M\_0C, M\_0D, vel, v, D, s, alpha} \right)
\]


2.

\[
\text{correct\_net} = \text{make\_HPN}(\text{net\_array})
\]

This also allow to verify if input data are consistent. If such is the case, then output data of function \text{make\_HPN} are saved in an array of cells called \text{correct\_net}; if input data are not consistent error messages are displayed. If data are consistent, in the prompt of Matlab the following information are visualized:

transition t1 --> msf=0 MFS=1
transition t2 --> msf=0 MFS=2
transition t3 --> server inf; weight 1;
transition t4 --> server inf; weight 1;
transition t5 --> server inf; weight 1;
transition t6 --> server inf; weight 1;

\[
\text{correct\_net} = \]

Columns 1 through 6

[6x7 double] [6x7 double] [5x1 double] [2x2 double] [1x4 double] [4x3 double]

Columns 7 through 8

[1x4 double] [1x4 double]

In particular, it is summarized the minimum firing speeds and the maximal firing speeds of continuous transitions, and the number of servers and the weight associated to discrete transitions. Then, it is visualized that \text{correct\_net} is an array of eight cells, and the dimension of each cell is explicitly shown. If \text{celldisp(correct\_net)} is executed then the following data are shown:

\[
\text{correct\_net}\{1\} = \text{Pre} = \begin{bmatrix}
0 & 1 & \text{NaN} & 0 & 0 & 0 & 0 \\
\text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} \\
0 & 0 & \text{NaN} & 0 & 1 & 0 & 0 \\
1 & 0 & \text{NaN} & 1 & 0 & 0 & 0 \\
0 & 0 & \text{NaN} & 0 & 0 & 0 & 1 \\
0 & 1 & \text{NaN} & 0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

\[
\text{correct\_net}\{2\} = \text{Post} = \begin{bmatrix}
1 & 0 & \text{NaN} & 0 & 0 & 0 & 0 \\
\text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} \\
0 & 0 & \text{NaN} & 1 & 0 & 0 & 0 \\
1 & 0 & \text{NaN} & 0 & 1 & 0 & 0 \\
0 & 0 & \text{NaN} & 0 & 0 & 1 & 0 \\
0 & 1 & \text{NaN} & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
correct_net{3} = M0 = [ 1
0
1
0
1 ];

correct_net{4} = vel = [ 0 1
0 2 ];

correct_net{5} = v = [ 1 2 1 2 ];

correct_net{6} = D = [ 2 NaN NaN
1.5 NaN NaN
1.5 NaN NaN ];

correct_net{7} = s = [ 0 0 0 0 ];

correct_net{8} = alpha = [ 1 1 1 1 ];

To simulate the net behavior we use function simulator_HPN, input data are correct_net plus three additional parameters that we assume as follows:

- \( time\_stop = 7 \) (the time length of simulation is assumed equal to seven)
- \( simulation\_type = 1 \) (specifies that we want visualized the evolution graph without interruptions until \( time\_stop \) is reached)
- \( J = [ 1 1 ] \) (the objective of function we want maximize for each Macro Period is equal to \( V_1 + V_2 \) where \( V_1 \) and \( V_2 \) are the IFS of continuous transitions \( t_1 \) and \( t_2 \))

Now, we execute

\[
Evol = simulator\_HPN (correct\_net, 7, 1, \ [ 1 1 ] )
\]

The evolution graph is generated on the screen:

The current continuous marking

\( 1 \)

Istantaneous firing speeds
\( t_1 \ 1 \ ) strongly enabled
\( t_2 \ 2 \ ) strongly enabled

The current discrete marking

\[ 0 \ 1 \ \ 0 \ 1 \]
Residual times of clocks $Q$

\[
t_3 \quad 2.000 \\
t_4 \quad \text{NaN} \\
t_5 \quad 1.500 \\
t_6 \quad \text{NaN}
\]

Enabling vector $e$

\[
1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0
\]

Place $p_1$ gets empty after a time interval 1.000 (time $\text{tot}=1$)
The current continuous marking

\[
0
\]

Istantaneous firing speeds

\[
t_1 \quad \text{1 strongly enabled} \\
t_2 \quad \text{1 weakly enabled}
\]

The current discrete marking

\[
0 \quad 1 \quad 0 \quad 1
\]

Residual times of clocks $Q$

\[
t_3 \quad 1.000 \\
t_4 \quad \text{NaN} \\
t_5 \quad 0.500 \\
t_6 \quad \text{NaN}
\]

Enabling vector $e$

\[
1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0
\]

The discrete transition that fire at time 0.500 (time $\text{tot}=1.500$) is $t_5$ (1 time)

The current continuous marking

\[
0
\]

Istantaneous firing speeds

\[
t_1 \quad \text{1 strongly enabled} \\
t_2 \quad \text{0 not enabled}
\]

The current discrete marking

\[
0 \quad 1 \quad 1 \quad 0
\]
Residual times of clocks $Q$
- $t_3$ 0.500
- $t_4$ NaN
- $t_5$ NaN
- $t_6$ 2.779

Enabling vector $e$

```
1 0 1 0 0 1
```

The discrete transition that fire at time 0.500 (time $tot=2.000$) is $t_3$ (1 time)

The current continuous marking

```
0.5000
```

Instantaneous firing speeds
- $t_1$ 0 not enabled
- $t_2$ 0 not enabled

The current discrete marking

```
1 0 1 0
```

Residual times of clocks $Q$
- $t_3$ NaN
- $t_4$ 2.929
- $t_5$ NaN
- $t_6$ 2.279

Enabling vector $e$

```
0 0 0 1 0 1
```

The discrete transition that fire at time 2.279 (time $tot=4.279$) is $t_6$ (1 time)

The current continuous marking

```
0.5000
```

Instantaneous firing speeds
- $t_1$ 0 not enabled
- $t_2$ 2 strongly enabled

The current discrete marking

```
1 0 0 1
```
Residual times of clocks Q

\begin{verbatim}
t3  NaN
t4  0.651
nt5  1.500
t6  NaN
\end{verbatim}

Enabling vector e

\begin{verbatim}
0  1  0  1  1  0
\end{verbatim}

Place p1 gets empty after a time interval 0.250 (time tot=4.528843e+000)

The current continuous marking

\begin{verbatim}
0
\end{verbatim}

Instantaneous firing speeds

\begin{verbatim}
t1  0  not enabled
t2  0  weakly enabled
\end{verbatim}

The current discrete marking

\begin{verbatim}
1  0  0  1
\end{verbatim}

Residual times of clocks Q

\begin{verbatim}
t3  NaN
t4  0.401
nt5  1.250
t6  NaN
\end{verbatim}

Enabling vector e

\begin{verbatim}
0  1  0  1  1  0
\end{verbatim}

The discrete transition that fire at time 0.401 (time tot=4.929) is t4 (1 time)

The current continuous marking

\begin{verbatim}
0
\end{verbatim}

Instantaneous firing speeds

\begin{verbatim}
t1  1  strongly enabled
t2  1  weakly enabled
\end{verbatim}

The current discrete marking

\begin{verbatim}
0  1  0  1
\end{verbatim}
Residual times of clocks $Q$

- $t_3 = 2.000$
- $t_4 = NaN$
- $t_5 = 0.849$
- $t_6 = NaN$

Enabling vector $e$

```
1 1 1 0 1 0
```

The discrete transition that fire at time 0.849 (time $tot=5.779$) is $t_5$ (1 time)

The current continuous marking

```
0
```

Instantaneous firing speeds

- $t_1$: 1 strongly enabled
- $t_2$: 0 not enabled

The current discrete marking

```
0 1 1 0
```

Residual times of clocks $Q$

- $t_3 = 1.151$
- $t_4 = NaN$
- $t_5 = NaN$
- $t_6 = 3.546$

Enabling vector $e$

```
1 0 1 0 0 1
```

The discrete transition that fire at time 1.151 (time $tot=6.929$) is $t_3$ (1 time)

The current continuous marking

```
1.1506
```

Instantaneous firing speeds

- $t_1$: 0 not enabled
- $t_2$: 0 not enabled

The current discrete marking
1 0 1 0

Residual times of clocks Q
\[t_3 \quad NaN\]
\[t_4 \quad 1.443\]
\[t_5 \quad NaN\]
\[t_6 \quad 2.396\]

Enabling vector e
\[
\begin{array}{cccccc}
0 & 0 & 0 & 1 & 0 & 1
\end{array}
\]

\[
\text{Evol} =
\]

Columns 1 through 6
\[
\begin{bmatrix}
3 & \text{[5x10 double]} & \text{[2x10 double]} & \text{[1x10 double]} & \text{[1x10 double]} & \text{[1x10 double]}
\end{bmatrix}
\]

Columns 7 through 10
\[
\begin{bmatrix}
\text{[1x10 double]} & \text{[1x10 double]} & \text{[1x10 cell]} & \text{[1x4 double]}
\end{bmatrix}
\]

Output data of function \texttt{simulator	extunderscore HPN} are saved in an array of then cells called \texttt{Evol} that provides the input of function \texttt{analysis	extunderscore HPN}. If we execute
\[
\text{analysed	extunderscore net} = \text{analysis	extunderscore HPN}(\text{Evol})
\]

the following data are shown in the screen:

\[
E[M(p1)]=0.367
\]
\[
E[M(p2)]=0.429
\]
\[
E[M(p3)]=0.571
\]
\[
E[M(p4)]=0.571
\]
\[
E[M(p5)]=0.429
\]
\[
\text{max}(p1)=1.151
\]
\[
\text{max}(p2)=1.000
\]
\[
\text{max}(p3)=1.000
\]
\[
\text{max}(p4)=1.000
\]
\[
\text{max}(p5)=1.000
\]
\[
\text{Prob}[M(p2)=0]=0.571
\]
\[
\text{Prob}[M(p2)=1]=0.429
\]
\[
\text{Prob}[M(p3)=0]=0.429
\]
\[
\text{Prob}[M(p3)=1]=0.571
\]
\[
\text{Prob}[M(p4)=0]=0.429
\]
\[
\text{Prob}[M(p4)=1]=0.571
\]
Prob[M(p5)=0]=0.571
Prob[M(p5)=1]=0.429
E[v1]=0.571
E[v2]=0.550
E[t3]=0.286
E[t4]=0.143
E[t5]=0.286
E[t6]=0.143

Md(1)=[1 0 1 0 1 ]
f(1)=0.14286

Md(2)=[0 0 1 0 1 ]
f(2)=0.19277

Md(3)=[0 0 1 1 0 ]
f(3)=0.23580

Md(4)=[1 1 0 1 0 ]
f(4)=0.32555

Md(5)=[1 1 0 0 1 ]
f(5)=0.03571

Md(6)=[0 1 0 0 1 ]
f(6)=0.34295

Md(7)=[1 1 0 1 0 ]
f(7)=0.01007

analysed_net =

[1x5 double] [1x5 double] [1x2 double] [1x6 double] [5x2 double] [] [6x7 double]

Now, let us the function analysis_HPN may also have different inputs that enable us to specify the plots we want visualize. The following examples clarify this.

•

analysed_net = analysis_HPN(Evol,1)

statistic_plot = 1: the maximum and the average value of the marking during the simulation are displayed (see Figure 2);
Figure 2: \( \text{analysed}_\text{net} = \text{analysis}_\text{HPN}(\text{Evol}, 1) \)

- \( \text{analysed}_\text{net} = \text{analysis}_\text{HPN}(\text{Evol}, [], 1) \)
  \( \text{graph} = 1 \): this parameter is only relative to discrete nets, thus it has no effect in our example regardless of the value it assumes;

- \( \text{analysed}_\text{net} = \text{analysis}_\text{HPN}(\text{Evol}, [], [], -1) \)
  \( \text{marking}_\text{plot} = -1 \): the marking evolution of all places is shown (see Figure 3) in two different figures. The first one shows the marking evolution of places \( p_1, p_2, p_3, p_4 \), and the second one shows the marking evolution of places \( p_5 \).

- \( \text{analysed}_\text{net} = \text{analysis}_\text{HPN}(\text{Evol}, [], [], [], 1) \)
  \( Td_\text{firing}_\text{plot} = 1 \): a graph showing the time instants at which discrete transition have fired is shown (see Figure 4);

- \( \text{analysed}_\text{net} = \text{analysis}_\text{HPN}(\text{Evol}, [], [], [], [1 3]) \)
  \( Up_\text{marking}_\text{plot} = [1 3] \): the marking evolution of places \( p_1 \) and \( p_3 \) are plotted in the same figure (see Figure 5);

- \( \text{analysed}_\text{net} = \text{analysis}_\text{HPN}(\text{Evol}, [], [], [], [], 1) \)
  \( Pd_\text{prob}_\text{plot} = 1 \): the frequency of having a given number of tokens during the simulation run is shown for each discrete place (see Figure 6);

- \( \text{analysed}_\text{net} = \text{analysis}_\text{HPN}(\text{Evol}, [], [], [], [], [], 1) \)
  \( Pd_\text{ave}_\text{t}_\text{plot} = 1 \): this parameter is only relative to discrete nets, thus it has no effect in our example regardless of the value it assumes;

\(^2\)By default if the number of places is greater than 4, then the graphs are shown in different figures.
The evolution of marking $M(p_1)$ wrt time

The evolution of marking $M(p_2)$ wrt time

The evolution of marking $M(p_3)$ wrt time

The evolution of marking $M(p_4)$ wrt time

The evolution of marking $M(p_5)$ wrt time

Figure 3: $analysed\_net = analysis\_HPN(Evol, [], [], -1)$
Figure 4: *analysed_net = analysis_HPN(Evol, [], [], [], 1)*

Figure 5: *analysed_net = analysis_HPN(Evol, [], [], [], [1 3])*
Figure 6: $analyzed_{net} = analysis_{HPN}(Evol, [], [], [], [], [], 1)$
Figure 7: $\textit{analysed.net} = \textit{analysis.HPN}(\textit{Evol}, [], [], [], [], [], [], [], -1)$

Figure 8: $\textit{analysed.net} = \textit{analysis.HPN}(\textit{Evol}, [], [], [], [], [], [], [], -1)$

- $\textit{analysed.net} = \textit{analysis.HPN}(\textit{Evol}, [], [], [], [], [], [], [], [], -1)$

  $\textit{IFS.plot} = -1$: the IFS of all continuous transitions are shown (see Figure 7);

- $\textit{analysed.net} = \textit{analysis.HPN}(\textit{Evol}, [], [], [], [], [], [], [], [], -1)$

  $\text{up.IFS.plot} = -1$: the IFS of all continuous transitions are shown in the same figure (see Figure 8);

- $\textit{analysed.net} = \textit{analysis.HPN}(\textit{Evol}, [], [], [], [], [], [], [], [], 1)$

  $\textit{IFS.ave.plot} = 1$: an histogram showing the average firing speed of continuous transitions is created (see Figure 9);

- $\textit{analysed.net} = \textit{analysis.HPN}(\textit{Evol}, [], [], [], [], [], [], [], [], 1)$

  $\text{Td.freq.plot} = 1$: an histogram showing the firing frequency of discrete transitions is created (see Figure 10);
Figure 9: \textit{analysed\_net} = \textit{analysis\_HPN}(\textit{Evol},[],[],[],[],[],[],[],[],1)

Figure 10: \textit{analysed\_net} = \textit{analysis\_HPN}(\textit{Evol},[],[],[],[],[],[],[],[],[],[],1)
References


