Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

NLR-TP-2006-689

Hybrid Petri nets with diffusion that have into-mappings with generalised stochastic hybrid processes

M.H.C. Everdij and H.A.P. Blom

This report contains a paper published in Stochastic Hybrid Systems - Theory and Safety Critical Applications, Editors: H.A.P. Blom and J. Lygeros, Published by Springer, July 2006.

Customer:	National Aerospace Laboratory NLR
Working Plan number:	2005 AT.1.A
Owner:	National Aerospace Laboratory NLR
Division:	Air Transport
Distribution:	Unlimited
Classification title:	Unclassified
	September 2006

Approved by:

Author	Reviewer	Managing department
ME '06	Anonymous peer reviewers	22/12/06

Hybrid Petri Nets with Diffusion that have Into-Mappings with Generalised Stochastic Hybrid Processes

Mariken H.C. Everdij[1] and Henk A.P. Blom[1]

National Aerospace Laboratory NLR, everdij@nlr.nl, blom@nlr.nl

Summary. Generalised Stochastic Hybrid Processes (GSHPs) are known as the largest class of Markov processes virtually describing all continuous-time processes including diffusion. In general, the state space of a GSHP is of hybrid type, i.e. a Kronecker product of a discrete set and a continuous-valued space. Since Stochastic Petri Nets have proven to be extremely useful in developing continuous-time Markov Chain models for complex practical discrete-valued processes, there is a clear need for a type of hybrid Petri Nets that can play a similar role for developing GSHP models for complex practical problems. To fulfil this need, the report defines a Stochastically and Dynamically Coloured Petri Net (SDCPN), and proves that there exist into-mappings between GSHPs and SDCPNs.

1 Introduction

Davis [6] has introduced Piecewise Deterministic Markov Processes (PDPs) as the most general class of continuous-time Markov processes which include both discrete and continuous processes, except diffusion. A PDP $\{\xi_t\}$ consists of two components: a piecewise constant component $\{\theta_t\}$ and a piecewise continuous valued component $\{x_t\}$, which follows the solution of a θ_t-dependent ordinary differential equation. A jump in $\{\xi_t\}$ occurs when $\{x_t\}$ hits the boundary of a predefined area, or according to a jump rate. If $\{x_t\}$ also makes a jump at a time when $\{\theta_t\}$ switches, this is said to be a hybrid jump.

Bujorianu et al [3] extended this PDP definition to Generalised Stochastic Hybrid Processes (GSHP) by including diffusion by means of Brownian motion. With this extension, between jumps, the process $\{x_t\}$ follows the solution of a θ_t-dependent stochastic (rather than ordinary) differential equation. GSHP forms a powerful and useful class of processes that have strong support in stochastic analysis and control.

A Petri Net is a bipartite graph of places (possible conditions or discrete modes) and transitions (possible mode switches). Tokens, which reside in the places, model which conditions or modes are current. Petri Nets, see e.g. [4], and their many extensions, see e.g. [5] for a good overview, have proven to be extremely useful in developing models for various complex practical applications. This usefulness is especially due to their specification power [4], which allows to develop a submodel for each entity of a complex operation, and next to combine the submodels in a constructive way. An example is Stochastic Petri Nets, which have been successfully used in developing continuous-time Markov Chain models for complex practical discrete-valued processes. For this reason, there is a clear need for a type of Petri Nets that can play a similar role for developing PDP or GSHP models for complex practical problems. Several hybrid state Petri Net extensions have been developed in the past. Main classes are:

- Hybrid Petri Net , [1]. Some places have a continuous amount of tokens that may be moved to other places by transitions.
- Fluid Stochastic Petri Net (FSPN), [16]. Some places have a continuous amount of tokens, the flow rate of which is influenced by the discrete part. The discrete part of the FSPN can be mapped to a continuous-time Markov chain.
- Extended Coloured Petri Net (ECPN), [17]. The token colours are real-valued vectors that may follow the solution path of a difference equation.
- High-Level Hybrid Petri Net (HLHPN), [12]. Again, the token colours are real-valued vectors that may follow the solution path of a difference equation, but in addition, a token switch between discrete places may generate a jump in the value of the real-valued vector.
- Differential Petri Nets , [8]. Differential places have a real-valued number of tokens and differential transitions fire with a certain speed that may also be negative.

For none of the above hybrid state Petri Nets it is clear how they relate to PDP. Moreover, none of them include Brownian motion as GSHP does. In order to improve this situation for PDP, Everdij and Blom [10], [11], developed a Petri Net extension named Dynamically Coloured Petri Net (DCPN) and proved that here exist into-mappings between PDPs and DCPNs. In [9], Everdij and Blom showed that this existence of into-mappings extends the power-hierarchy among various model types established by [14], [15]. This is shown in Figure 1, in which the well-known dependability models Reliability Block Diagrams and Fault Trees are at the basis of the hierarchy.

Although PDP form a very general class of continuous-time Markov processes which include both discrete and continuous processes, PDP do not include diffusion. The aim of the current chapter aims to solve this issue by

- including a diffusion term into the PDP definition, following [3], and referred to as GSHP (Generalised Stochastic Hybrid Process);
- introducing an extension of DCPN, referred to as Stochastically and Dynamically Coloured Petri Net (SDCPN), which also covers diffusion;
- and showing that there exist into-mappings between GSHP and SDCPN.

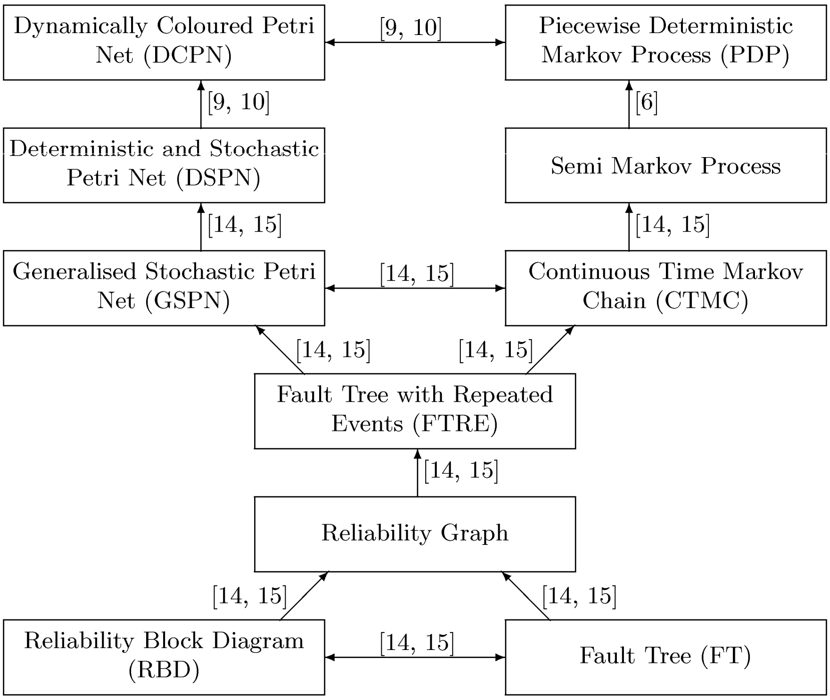


Fig. 1. Power hierarchy among various model types established by [6], [9], [10], [14], and [15]. An arrow from a model to another model indicates that the second model has more modelling power than the first model

The existence of such into-mappings allows combining the specification power of Petri Nets with the stochastic analysis and control power of GSHP. In addition, the into-mappings extend the power hierarchy of Figure 1 with GSHP and with GSHP-related Petri Nets.

The organisation of the paper is as follows. Section 2 briefly describes GSHP. Section 3 defines SDCPN. Section 4 shows that each GSHP can be represented by a SDCPN process. Section 5 shows that each SDCPN process can be represented by a GSHP. Section 6 presents a SDCPN model for a simple aircraft evolution example and its mapping to a GSHP. Section 7 draws conclusions.

2 Generalised Stochastic Hybrid Process

This section presents a definition of Generalised Stochastic Hybrid System (GSHS) and its GSHP solution, see [3]. As much as possible, the notation introduced by Davis [7] for Piecewise Deterministic Markov Process is used.

Definition 1. *A Generalised Stochastic Hybrid System (GSHS) is a nine-tuple (*$\mathbf{K}$*,* $d(\theta)$*,* x_0*,* θ_0*,* ∂E_θ*,* g_θ*,* g_θ^w*,* λ*,* Q*), together with some conditions* C_1 *–* C_4*.*

Below, first the structure of the elements in the tuple and the GSHS conditions are given, next the GSHS execution is explained.

2.1 GSHS Elements

The GSHS elements are defined as follows:

1. $\mathbf{K}$ is a countable set of discrete variables.
2. d is a map from $\mathbf{K}$ into $\mathbb{N}$, giving the dimensions of the continuous state process.
3. For each $\theta \in \mathbf{K}$, E_θ is an open subset of $\mathbb{R}^{d(\theta)}$, and ∂E_θ is its boundary.
4. θ_0 is an initial value in $\mathbf{K}$.
5. x_0 is an initial value in E_{θ_0}.
6. $g_\theta : \mathbb{R}^{d(\theta)} \to \mathbb{R}^{d(\theta)}$ is a vector field.
7. $g_\theta^w : \mathbb{R}^{d(\theta)} \to \mathbb{R}^{d(\theta)} \times \mathbb{R}^b$ is a matrix, with $b \in \mathbb{N}$.
8. $\lambda : E \to \mathbb{R}^+$ is a jump rate function, with $E = \cup_\theta E_\theta$.
9. $Q : E \cup \Gamma^* \to [0,1]$ is a probability measure, with $E = \cup_\theta E_\theta$ and Γ^* the reachable boundary of E.

2.2 GSHS Conditions

Following [3] (Assumptions 1, 2 and 3), the GSHS conditions are:

C_1 g_θ and g_θ^w are such[1] that for each initial state (θ, x) at initial time τ there exists a pathwise unique solution $x_t = \phi_{\theta,x,t-\tau}$ to $dx_t = g_\theta(x_t)dt + g_\theta^w(x_t)dw_t$, where $\{w_t\}$ is b-dimensional standard Brownian motion. If $t_\infty(\theta, x)$ denotes the explosion time of the flow $\phi_{\theta,x,t-\tau}$, i.e. $|\phi_{\theta,x,t-\tau}| \to \infty$ as $t \uparrow t_\infty(\theta, x)$, then it is assumed that $t_\infty(\theta, x) = \infty$ whenever $t_*(\theta, x) = \infty$. In other words, explosions are ruled out.

C_2 With $E = \cup_\theta E_\theta$, $\lambda : E \to \mathbb{R}^+$ is a measurable function such that for all $\xi \in E$, there is $\epsilon(\xi) > 0$ such that $t \to \lambda(\theta, \phi_{\theta,x,t})$ is integrable on $[0, \epsilon(\xi)[$.

C_3 With E as above and Γ^* the reachable boundary of E, Q maps $E \cup \Gamma^*$ into the set of probability measures on $(E, \mathcal{E})$, with $\mathcal{E}$ the Borel-measurable subsets of E, while for each fixed $A \in \mathcal{E}$, the map $\xi \to Q(A; \xi)$ is measurable and $Q(\{\xi\}; \xi) = 0$.

C_4 If $N_t = \sum_k I_{(t \geq \tau_k)}$, then it is assumed that for every starting point ξ and for all $t \in \mathbb{R}^+$, $\mathbb{E}N_t < \infty$. This means, there will be a finite number of jumps in finite time.

---

[1] [3] assumes Lipschitz continuity and boundedness.

2.3 GSHS Execution

The execution of a GSHS generates a Generalised Stochastic Hybrid Process (GSHP) $\{\xi_t\}$, with $\xi_t = (\theta_t, x_t)$, as follows:

For each $\theta \in \mathbf{K}$, consider the stochastic differential equation $dx_t = g_\theta(x_t)dt + g_\theta^w(x_t)dw_t$, where $\{w_t\}$ is b-dimensional standard Brownian motion. Given an initial value $x \in E_\theta$, under GSHS condition C_1, this differential equation has a pathwise unique solution. This means that if at some time instant τ the GSHP state assumes value $\xi_\tau = (\theta_\tau, x_\tau)$, then, as long as no jumps occur, the GSHP state at $t \geq \tau$ is given by $\xi_t = (\theta_t, x_t) = (\theta_\tau, \phi_{\theta_\tau, x_\tau, t-\tau})$, with $\phi_{\theta_\tau, x_\tau, t-\tau} = \int_\tau^t g_{\theta_s}(x_s)ds + \int_\tau^t g_{\theta_s}^w(x_s)dw_s$. At some moment in time, however, the GSHP state value may jump. Such moment is generated by either one of the following events, depending on which event occurs first:

1. A Poisson point process with jump rate $\lambda(\theta_t, x_t)$, $t > \tau$ generates a point.
2. The piecewise continuous process x_t is about to hit the boundary ∂E_{θ_τ} of E_{θ_τ}, $t > \tau$.

At the moment when either of these events occurs, the GSHP state makes a jump. The value of the GSHP state right after the jump is generated by using a transition measure Q, which is the probability measure of the GSHP state after the jump, given the value of the GSHP state immediately before the jump. After this, the GSHP state ξ_t evolves in a similar way from the new value onwards.

The GSHP process is generated by executing a GSHS through time as follows: Suppose at time $\tau_0 \triangleq 0$ the GSHP initial state is $\xi_0 = (\theta_0, x_0)$, then, if no jumps occur, the process state at $t \geq \tau_0$ is given by $\xi_t = (\theta_t, x_t) = (\theta_0, \phi_{\theta_0, x_0, t-\tau_0})$. The complementary distribution function for the time of the first jump (i.e. the probability that the first jump occurs at least $t - \tau_0$ time units after τ_0), also named the survivor function of the first jump, is then given by:

$$G_{\xi_0, t-\tau_0} \triangleq I_{(t-\tau_0 < t_*(\theta_0, x_0))} \cdot \exp\left\{ -\int_{\tau_0}^t \lambda(\theta_0, \phi_{\theta_0, x_0, s-\tau_0})ds \right\}, \qquad (1)$$

where I is an indicator function and $t_*(\theta_0, x_0)$ denotes the time until the first boundary hit after $t = \tau_0$, which is given by $t_*(\theta_0, x_0) \triangleq \inf\{t - \tau_0 > 0 \mid \phi_{\theta_0, x_0, t-\tau_0} \in \partial E_{\theta_0}\}$. The first factor in Equation (1) is explained by the boundary hitting process: after the process state has hit the boundary, which is when $t - \tau_0 = t_*(\theta_0, x_0)$, this first factor ensures that the survivor function evaluates to zero. The second factor in Equation (1) comes from the Poisson process: this second factor ensures that a jump is generated after an exponentially distributed time with a rate λ that is dependent on the GSHP state.

The time τ_1 until the first jump after τ_0 is generated by drawing a sample from a uniform distribution on $[0, 1]$, and then using a transformation that takes G into account. More formally (see [7], Section 23), the Hilbert cube $\Omega^H = \prod_{i=1}^{\infty} Y_i$, with Y_i a copy of $Y = [0,1]$, provides the canonical space for a countable sequence of independent random variables U_1, U_2, ..., each having uniform $[0,1]$ distribution, defined by $U_i(\omega) = \omega_i$ for elements $\omega = (\omega_1, \omega_2, \ldots) \in \Omega^H$. The complete probability space is $(\Omega, \mathfrak{F}, \mathfrak{P}, \{\mathfrak{F}_t\})$, with $\Omega = \Omega^H \times \Omega^B$, and where Ω^B supports the Brownian motion. Now, define

$$\psi_1(u, \xi_0, \omega) = \begin{cases} \inf\{t : G_{\xi_0, t-\tau_0}(\omega) \leq u\} \\ +\infty \text{ if the above set is empty} \end{cases}$$

and define $\sigma_1(\omega) = \tau_1(\omega) = \psi_1(U_1(\omega), \xi_0, \omega)$, then τ_1 is the time until the first jump.

The value of the hybrid process state to which the jump is made is generated by using the transition measure Q, which is the probability measure of the hybrid state after the jump, given the value of the hybrid state immediately before the jump. The Hilbert cube from above is again used: Let $\psi_2 : [0,1] \times (E \cup \Gamma^*) \to E$, with $E = \cup_\theta E_\theta$ and Γ^* the reachable boundary of E, be a measurable function such that $l\{u : \psi_2(u,\xi) \in B\} = Q(B,\xi)$ for B Borel measurable. Then $\xi_{\tau_1} = \psi_2(U_2(\omega), \xi)$ is a sample from $Q(\cdot, \xi)$.

With this, the algorithm to determine a sample path for the hybrid state process ξ_t, $t \geq 0$, from the initial state $\xi_0 = (\theta_0, x_0)$ on, is in two iterative steps; define $\tau_0 \triangleq 0$ and let for $k = 0$, $\xi_{\tau_k} = (\theta_{\tau_k}, x_{\tau_k})$ be the initial state, then for $k = 1, 2, \ldots$:

Step 1: Draw a sample σ_k from survivor function $G_{\xi_{\tau_{k-1}}, t-\tau_{k-1}}(\omega)$, i.e. $\sigma_k(\omega) = \psi_1(U_{2k-1}(\omega), \xi_{\tau_{k-1}}, \omega)$. Then the time τ_k of the kth jump is $\tau_k = \tau_{k-1} + \sigma_k$. The sample path up to the kth jump is given by

$$\xi_t = (\theta_{\tau_{k-1}}, \phi_{\theta_{\tau_{k-1}}, x_{\tau_{k-1}}, t-\tau_{k-1}}), \quad \tau_{k-1} \leq t < \tau_k \text{ and } \tau_k \leq \infty.$$

Step 2: Draw a multi-dimensional sample ζ_k from transition measure $Q(\cdot; \xi'_{\tau_k})$, where $\xi'_{\tau_k} = (\theta_{\tau_{k-1}}, \phi_{\theta_{\tau_{k-1}}, x_{\tau_{k-1}}, \tau_k - \tau_{k-1}})$, i.e. $\zeta_k = \psi_2(U_{2k}(\omega), \xi'_{\tau_k})$. Then, if $\tau_k < \infty$, the process state at the time τ_k of the kth jump is given by

$$\xi_{\tau_k} = \zeta_k.$$

3 Stochastically and Dynamically Coloured Petri Net (SDCPN)

This section presents a definition of Stochastically and Dynamically Coloured Petri Net (SDCPN). As much as possible, the notation introduced by Jensen [13] for Coloured Petri Net is used.

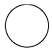
Definition 2. *A Stochastically and Dynamically Coloured Petri Net (SDCPN) is a 12-tuple* SDCPN $= (\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{C}, \mathcal{V}, \mathcal{W}, \mathcal{G}, \mathcal{D}, \mathcal{F}, \mathcal{I})$*, together with some rules* R_0 – R_4.

Below, first the structure of the elements in the tuple is given, next the SDCPN evolution through time is explained, finally, the SDCPN generated process is outlined.

3.1 SDCPN Elements

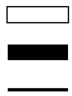
The SDCPN elements are defined as follows:

1. $\mathcal{P}$ is a finite set of places. In a graphical notation, places are denoted by circles:

   Place: 

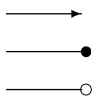
2. $\mathcal{T}$ is a finite set of transitions, such that $\mathcal{T} \cap \mathcal{P} = \emptyset$. The set $\mathcal{T}$ consists of 1) a set $\mathcal{T}_G$ of guard transitions, 2) a set $\mathcal{T}_D$ of delay transitions and 3) a set $\mathcal{T}_I$ of immediate transitions, with $\mathcal{T} = \mathcal{T}_G \cup \mathcal{T}_D \cup \mathcal{T}_I$, and $\mathcal{T}_G \cap \mathcal{T}_D = \mathcal{T}_D \cap \mathcal{T}_I = \mathcal{T}_I \cap \mathcal{T}_G = \emptyset$. Notations are:

   Guard transition:
   Delay transition:
   Immediate transition:

3. $\mathcal{A}$ is a finite set of arcs such that $\mathcal{A} \cap \mathcal{P} = \mathcal{A} \cap \mathcal{T} = \emptyset$. The set $\mathcal{A}$ consists of 1) a set $\mathcal{A}_O$ of ordinary arcs, 2) a set $\mathcal{A}_E$ of enabling arcs and 3) a set $\mathcal{A}_I$ of inhibitor arcs, with $\mathcal{A} = \mathcal{A}_O \cup \mathcal{A}_E \cup \mathcal{A}_I$, and $\mathcal{A}_O \cap \mathcal{A}_E = \mathcal{A}_E \cap \mathcal{A}_I = \mathcal{A}_I \cap \mathcal{A}_O = \emptyset$. Notations are:

   Ordinary arc:
   Enabling arc:
   Inhibitor arc:

4. $\mathcal{N} : \mathcal{A} \to \mathcal{P} \times \mathcal{T} \cup \mathcal{T} \times \mathcal{P}$ is a node function which maps each arc A in $\mathcal{A}$ to a pair of ordered nodes $\mathcal{N}(A)$. The place of $\mathcal{N}(A)$ is denoted by $P(A)$, the transition of $\mathcal{N}(A)$ is denoted by $T(A)$, such that for all $A \in \mathcal{A}_E \cup \mathcal{A}_I$: $\mathcal{N}(A) = (P(A), T(A))$ and for all $A \in \mathcal{A}_O$: either $\mathcal{N}(A) = (P(A), T(A))$ or $\mathcal{N}(A) = (T(A), P(A))$. Further notation:
   - $A(T) = \{A \in \mathcal{A} \mid T(A) = T\}$ denotes the set of arcs connected to transition T, with $A(T) = A_{in}(T) \cup A_{out}(T)$, where
   - $A_{in}(T) = \{A \in A(T) \mid \mathcal{N}(A) = (P(A), T)\}$ is the set of input arcs of T and
   - $A_{out}(T) = \{A \in A(T) \mid \mathcal{N}(A) = (T, P(A))\}$ is the set of output arcs of T. Moreover,
   - $A_{in,O}(T) = A_{in}(T) \cap \mathcal{A}_O$ is the set of ordinary input arcs of T,
   - $A_{in,OE}(T) = A_{in}(T) \cap \{\mathcal{A}_E \cup \mathcal{A}_O\}$ is the set of input arcs of T that are either ordinary or enabling, and
   - $P(A(T))$ is the set of places connected to T by the set of arcs $A(T)$.

Finally, $\{A_i \in \mathcal{A}_I \mid \exists A \in \mathcal{A}, A \neq A_i : \mathcal{N}(A) = \mathcal{N}(A_i)\} = \emptyset$, i.e., if an inhibitor arc points from a place P to a transition T, there is no other arc from P to T.

5. $\mathcal{S}$ is a finite set of colour types. Each colour type is to be written in the form $\mathbb{R}^n$, with n a natural number and with $\mathbb{R}^0 = \emptyset$.
6. $\mathcal{C} : \mathcal{P} \to \mathcal{S}$ is a colour function which maps each place $P \in \mathcal{P}$ to a specific colour type in $\mathcal{S}$.
7. $\mathcal{I} : \mathcal{P} \to \mathcal{C}(\mathcal{P})_{ms}$ is an initialisation function, where $\mathcal{C}(P)_{ms}$ for $P \in \mathcal{P}$ denotes the set of all multisets over $\mathcal{C}(P)$. It defines the initial marking of the net, i.e., for each place it specifies the number of tokens (possibly zero) initially in it, together with the colours they have, and their ordering per place.
8. $\mathcal{V}$ is set of a token colour functions. For each place $P \in \mathcal{P}$ for which $\mathcal{C}(P) \neq \mathbb{R}^0$, it contains a function $\mathcal{V}_P : \mathcal{C}(P) \to \mathcal{C}(P)$ which satisfies conditions that ensure a pathwise unique solution.
9. $\mathcal{W}$ is set of a token colour matrix functions. For each place $P \in \mathcal{P}$ for which $\mathcal{C}(P) \neq \mathbb{R}^0$, it contains a function $\mathcal{W}_P : \mathcal{C}(P) \to \mathcal{C}(P) \times \mathcal{C}'(P)$, which satisfies conditions that ensure a pathwise unique solution, and where $\mathcal{C}'(P)$ collects the Brownian motion terms. Here, $\mathcal{C}'$ maps $\mathcal{P}$ into $\mathbb{R}^b$, with $b \in \mathbb{R}$ a constant.
10. $\mathcal{G}$ is a set of transition guards. For each $T \in \mathcal{T}_G$, it contains a transition guard $\mathcal{G}_T : \mathcal{C}(P(A_{in,OE}(T))) \to \{\text{True, False}\}$. $\mathcal{G}_T(c)$ evaluates to True if c is in the boundary ∂G_T of an open subset G_T in $\mathcal{C}(P(A_{in,OE}(T)))$. Here, if $P(A_{in,OE}(T))$ contains more than one place, e.g., $P(A_{in,OE}(T)) = \{P_i, \ldots, P_j\}$, then $\mathcal{C}(P(A_{in,OE}(T)))$ is defined by $\mathcal{C}(P_i) \times \cdots \times \mathcal{C}(P_j)$. If $\mathcal{C}(P(A_{in,OE}(T))) = \mathbb{R}^0$ then $\partial G_T = \emptyset$ and the guard will always evaluate to False.
11. $\mathcal{D}$ is a set of transition enabling rate functions. For each $T \in \mathcal{T}_D$, it contains an integrable transition enabling rate function $\delta_T : \mathcal{C}(P(A_{in,OE}(T))) \to \mathbb{R}_0^+$, which, if T is evaluated from stopping time τ on, specifies a delay time equal to $\mathcal{D}_T(\tau) = \inf\{t \mid e^{-\int_\tau^t \delta_T(c_s)ds} \leq u\}$, where u is a random number drawn from $U[0,1]$ at τ. If $\mathcal{C}(P(A_{in,OE}(T))) = \mathbb{R}^0$ then δ_T is a constant function.
12. $\mathcal{F}$ is a set of firing measures. For each $T \in \mathcal{T}$ it specifies a probability measure $\mathcal{F}_T$ which maps $\mathcal{C}(P(A_{in,OE}(T)))$ into the set of probability measures on $\{0,1\}^{|A_{out}(T)|} \times \mathcal{C}(P(A_{out}(T)))$.

3.2 SDCPN Execution

The execution of a SDCPN provides a series of increasing stopping times, $\tau_0 < \tau_i < \tau_{i+1}$, with for $t \in (\tau_i, \tau_{i+1})$ a fixed number of tokens per place and per token a colour which is the solution of a stochastic differential equation. This number of tokens and the colours of these tokens are generated as follows:

Each token residing in place P has a colour of type $\mathcal{C}(P)$. If a token in place P has colour c at time τ, and if it remains in that place up to time

$t > \tau$, then the colour c_t at time t equals the unique solution of the stochastic differential equation $dc_t = \mathcal{V}_P(c_t)dt + \mathcal{W}_P(c_t)dw_t$ with initial condition $c_\tau = c$.

A transition T is *pre-enabled* if it has at least one token per incoming ordinary and enabling arc in each of its input places and has no token in places to which it is connected by an inhibitor arc; denote $\tau_1^{pre} = \inf\{t \mid T \text{ is pre-enabled at time } t\}$. Consider one token per ordinary and enabling arc in the input places of T and write $c_t \in \mathcal{C}(P(A_{in,OE}(T)))$, $t \geq \tau_1^{pre}$, as the column vector containing the colours of these tokens; c_t may change through time according to its corresponding token colour functions. If this vector is not unique (for example, one input place contains several tokens per arc), all possible such vectors are executed in parallel.

A transition T is *enabled* if it is pre-enabled and a second requirement holds true. For $T \in \mathcal{T}_I$, the second requirement automatically holds true. For $T \in \mathcal{T}_G$, the second requirement holds true when $\mathcal{G}_T(c_t) = \text{True}$. For $T \in \mathcal{T}_D$, the second requirement holds true $\mathcal{D}_T(\tau_1^{pre})$ units after τ_1^{pre}. Guard or delay evaluation of a transition T stops when T is not pre-enabled anymore, and is restarted when it is.

For the evaluation of $\mathcal{D}_T(\tau_1^{pre})$, use is made of a Hilbert cube $\Omega^H = \prod_{i=1}^{\infty} Y_i$, with Y_i a copy of $Y = [0,1]$, which provides the canonical space for a countable sequence of independent random variables U_1, U_2, ..., each having a uniform $[0,1]$ distribution, defined by $U_i(\omega) = \omega_i$ for elements $\omega = (\omega_1, \omega_2, \ldots) \in \Omega^H$. This Hilbert cube applies as follows: Suppose T is a delay transition that is pre-enabled at time τ and has vector of input colours c_t at time $t \geq \tau$. Then transition T is enabled at random time $\inf\{t : \exp\left\{-\int_\tau^t \delta_T(c_s)ds\right\} \leq U_i\}$, with $\inf\{\ \} = +\infty$. The complete probability space is $(\Omega, \mathfrak{F}, \mathfrak{P}, \{\mathfrak{F}_t\})$, with $\Omega = \Omega^H \times \Omega^B$, and where Ω^B supports the Brownian motion.

In case of competing enablings, the following rules apply:

- R_0 The firing of an immediate transition has priority over the firing of a guard or a delay transition.
- R_1 If one transition becomes enabled by two or more disjoint sets of input tokens at exactly the same time, then it will fire these sets of tokens independently, at the same time.
- R_2 If one transition becomes enabled by two or more non-disjoint sets of input tokens at exactly the same time, then the set that is fired is selected randomly.
- R_3 If two or more transitions become enabled at exactly the same time by disjoint sets of input tokens, then they will fire at the same time.
- R_4 If two or more transitions become enabled at exactly the same time by non-disjoint sets of input tokens, then the transition that will fire is selected randomly.

Here, two sets of input tokens are disjoint if they have no tokens in common that are reserved by ordinary arcs, i.e., they may have tokens in common that are reserved by enabling arcs.

If T is enabled, suppose this occurs at time τ_1, it removes one token per arc in $A_{in,O}(T)$ from each of its input places. At this time τ_1, T produces zero or one token along each output arc: If c_{τ_1} is the vector of colours of tokens that enabled T and (f, a_{τ_1}) is a sample from $\mathcal{F}_T(\cdot; c_{\tau_1})$, then vector f specifies along which of the output arcs of T a token is produced (f holds a one at the corresponding vector components and a zero at the arcs along which no token is produced) and a_{τ_1} specifies the colours of the produced tokens. The colours of the new tokens have sample paths that start at time τ_1.

For drawing the sample from $\mathcal{F}_T(\cdot; c_{\tau_1})$, again use is made of the Hilbert cube Ω^H: Let $\psi_2^T : [0,1] \times \mathcal{C}(P(A_{in,OE}(T))) \to \{0,1\}^{|A_{out}(T)|} \times \mathcal{C}(P(A_{out}(T)))$ be a measurable function such that $l\{u : \psi_2^T(u,c) \in B\} = \mathcal{F}_T(B,c)$ for B in the Borel set of $\{0,1\}^{|A_{out}(T)|} \times \mathcal{C}(P(A_{out}(T)))$. Then a sample from $\mathcal{F}_T(\cdot; c_{\tau_1})$ is given by $\psi_2^T(U_2(\omega), c_{\tau_1})$, if c_{τ_1} is the vector of input colours that enabled T.

In order to keep track of the identity of individual tokens, the tokens in a place are ordered according to the time at which they entered the place, or, if several tokens are produced for one place at the same time, according to the order within the set of arcs $\mathcal{A} = \{A_1, \ldots, A_{|\mathcal{A}|}\}$ along which these tokens were produced (the firing measure produces zero or one token along each output arc).

3.3 SDCPN Stochastic Process

The SDCPN generates a stochastic process which is uniquely defined as follows: The process state at time t is defined by the numbers of tokens in each place, and the colours of these tokens. Provided there is a unique ordering of SDCPN places, and a unique ordering of tokens within a place, this characterisation is unique, except at time instants when one or more transitions fire. To make this characterisation of SDCPN process state unique, it is defined as follows:

- At times t when no transition fires, the number of tokens in each place is uniquely characterised by the vector $(v_{1,t}, \ldots, v_{|\mathcal{P}|,t})$ of length $|\mathcal{P}|$, where $v_{i,t}$ denotes the number of tokens in place P_i at time t and $\{1, \ldots, |\mathcal{P}|\}$ refers to a unique ordering of places adopted for SDCPN. At time instants when one or more transitions fire, uniqueness of $(v_{1,t}, \ldots, v_{|\mathcal{P}|,t})$ is assured as follows: Suppose that τ is such time instant at which one transition or a sequence of transitions fires. Next, assume without loss of generality, that this sequence of transitions is $\{T_1, T_2, \ldots, T_m\}$ and that time is running again after T_m (note that T_1 must be a guard or a delay transition, and T_2 through T_m must be immediate transitions). Then the number of tokens in each place at time t is defined as that vector $(v_{1,t}, \ldots, v_{|\mathcal{P}|,t})$ that occurs after T_m has fired. This construction also ensures that the process $(v_{1,t}, \ldots, v_{|\mathcal{P}|,t})$ has limits from the left and is continuous from the right, i.e., it satisfies the càdlàg property.

- If $(v_{1,t}, \dots, v_{|\mathcal{P}|,t})$ is the distribution of the tokens among the places of the SDCPN at time t, which is uniquely defined above, then the associated colours of these tokens are uniquely gathered in a vector as follows: This vector first contains all colours of tokens in place P_1, next all colours of tokens in place P_2, etc, until place $P_{|\mathcal{P}|}$, where $\{1, \dots, |\mathcal{P}|\}$ refers to a unique ordering of places adopted for SDCPN. Within a place the colours of the tokens are ordered according to the unique ordering of tokens within their place defined for SDCPN (see under SDCPN execution above). Since $(v_{1,t}, \dots, v_{|\mathcal{P}|,t})$ satisfies the càdlàg property, the corresponding vector of token colours does too. An additional case occurs, however, when $(v_{1,t}, \dots, v_{|\mathcal{P}|,t})$ jumps to the same value again, so that only the process associated with the vector of token colours makes a jump at time τ. In that case, let the process associated with the vector of token colours be defined according to the timing construction as described for $(v_{1,t}, \dots, v_{|\mathcal{P}|,t})$ above (i.e. at time τ, the process associated with the vector of token colours is defined as that vector of token colours that occurs after the last transition has fired in the sequence of transitions that fire at time τ).

  With this, the SDCPN definition is complete.

4 Generalised Stochastic Hybrid Processes into Stochastically and Dynamically Coloured Petri Nets

This section shows that each Generalised Stochastic Hybrid Process can be represented by a Stochastically and Dynamically Coloured Petri Net, by providing a pathwise equivalent into-mapping from GSHP into the set of SDCPN processes.

Theorem 1. *For any arbitrary Generalised Stochastic Hybrid Process with a finite domain* $\mathbf{K}$ *there exists P-almost surely a pathwise equivalent process generated by a Stochastically and Dynamically Coloured Petri Net* ($\mathcal{P}$, $\mathcal{T}$, $\mathcal{A}$, $\mathcal{N}$, $\mathcal{S}$, $\mathcal{C}$, $\mathcal{I}$, $\mathcal{V}$, $\mathcal{W}$, $\mathcal{G}$, $\mathcal{D}$, $\mathcal{F}$) *satisfying* R_0 *through* R_4.

Proof. Consider an arbitrary GSHP $\{\theta_t, x_t\}$ described by the GSHS elements $\{\mathbf{K}, d(\theta), x_0, \theta_0, \partial E_\theta, g_\theta, \lambda, Q\}$.

First, we construct a SDCPN, the elements $\{\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{C}, \mathcal{I}, \mathcal{V}, \mathcal{W}, \mathcal{G}, \mathcal{D}, \mathcal{F}\}$ and the rules R_0 – R_4 of which are characterised in terms of the GSHS elements $\{\mathbf{K}, d(\theta), x_0, \theta_0, \partial E_\theta, g_\theta, \lambda, Q\}$ as follows:

$\mathcal{P} = \{P_\theta; \theta \in \mathbf{K}\}$. Hence, for each $\theta \in \mathbf{K}$ there is one place P_θ.

$\mathcal{T} = \mathcal{T}_G \cup \mathcal{T}_D \cup \mathcal{T}_I$, with $\mathcal{T}_I = \emptyset$, $\mathcal{T}_G = \{T^G_\theta; \theta \in \mathbf{K}\}$, $\mathcal{T}_D = \{T^D_\theta; \theta \in \mathbf{K}\}$. Hence, for each place P_θ there is one guard transition T^G_θ and one delay transition T^D_θ.

$\mathcal{A} = \mathcal{A}_O \cup \mathcal{A}_E \cup \mathcal{A}_I$, with $|\mathcal{A}_I| = 0$, $|\mathcal{A}_E| = 0$, and $|\mathcal{A}_O| = 2|\mathbf{K}| + 2|\mathbf{K}|^2$.

$\mathcal{N}$: The node function maps each arc in $\mathcal{A} = \mathcal{A}_O$ to a pair of nodes. These connected pairs of nodes are: $\{(P_\theta, T_\theta^G); \theta \in \mathbf{K}\} \cup \{(P_\theta, T_\theta^D); \theta \in \mathbf{K}\} \cup \{(T_\theta^G, P_\vartheta); \theta, \vartheta \in \mathbf{K}\} \cup (T_\theta^D, P_\vartheta); \theta, \vartheta \in \mathbf{K}\}$. Hence, each place P_θ has two outgoing arcs: one to guard transition T_θ^G and one to delay transition T_θ^D. Each transition has $|\mathbf{K}|$ outgoing arcs: one arc to each place in $\mathcal{P}$.

$\mathcal{S} = \{\mathbb{R}^{d(\theta)}; \theta \in \mathbf{K}\}$.

$\mathcal{C}$: For all $\theta \in \mathbf{K}$, $\mathcal{C}(P_\theta) = \mathbb{R}^{d(\theta)}$.

$\mathcal{I}$: Place P_{θ_0} contains one token with colour x_0. All other places initially contain zero tokens.

$\mathcal{V}$: For all $\theta \in \mathbf{K}$, $\mathcal{V}_{P_\theta}(\cdot) = g_\theta(\cdot)$.

$\mathcal{W}$: For all $\theta \in \mathbf{K}$, $\mathcal{W}_{P_\theta}(\cdot) = g_\theta^w(\cdot)$.

$\mathcal{G}$: For all $\theta \in \mathbf{K}$, $\partial G_{T_\theta^G} = \partial E_\theta$.

$\mathcal{D}$: For all $\theta \in \mathbf{K}$, $\delta_{T_\theta^D}(\cdot) = \lambda(\theta, \cdot)$. Moreover, for the evaluation of the SDCPN survivor functions, the same Hilbert cube applies as the one applied by the GSHP.

$\mathcal{F}$: If x denotes the colour of the token removed from place P_θ, $(\theta \in \mathbf{K})$, at the transition firing, then for all $\vartheta' \in \mathbf{K}$, $x' \in E_{\vartheta'}$: $\mathcal{F}_{T_\theta^G}(e', x'; x) = Q(\vartheta', x'; \theta, x)$, where e' is the vector of length $|\mathbf{K}|$ containing a one at the component corresponding with arc $(T_\theta^G, P_{\vartheta'})$ and zeros elsewhere. For all $\theta \in \mathbf{K}$, $\mathcal{F}_{T_\theta^D} = \mathcal{F}_{T_\theta^G}$. Moreover, for the evaluation of the SDCPN firing, the same Hilbert cube applies as the one applied by the GSHP.

R_0 – R_4: Since there are no immediate transitions in the constructed SDCPN instantiation, rule R_0 holds true. Since there is only one token in the constructed SDCPN instantiation, R_1 – R_3 also hold true. Rule R_4 is in effect when for particular θ, transitions T_θ^G and T_θ^D become enabled at exactly the same time. Since λ is integrable, the probability that this occurs is zero, yielding that R_4 holds with probability one. However, if this event should occur, then due to the fact that the firing measures for the guard transition and the delay transition are equal, the application of rule R_4 has no effect on the path of the SDCPN process.

This shows that for any GSHS we are able to construct a SDCPN instantiation. Next, we have to show that the SDCPN execution delivers the 'same' cadlag stochastic process as the GSHS execution does.

In the SDCPN instantiation constructed, initially there is one token in place P_{θ_0}. Because each transition firing removes one token and produces one token, the number of tokens does not change for $t > 0$. Hence, for $t > 0$ there is one token and the possible places for this single token are $\{P_\vartheta; \vartheta \in \mathbf{K}\}$. Figure 2 shows the situation at some time τ_{k-1}, when the GSHP is given by $(\theta_{\tau_{k-1}}, x_{\tau_{k-1}})$. The token resides in place P_{ϑ_i}, which models that $\theta_{\tau_{k-1}} = \vartheta_i$. This token has colour $x_{\tau_{k-1}}$. The colour of the token up to and at the time of the next jump is evaluated according to two steps that are similar to those of GSHP:

Step 1: While the token is residing in place P_{ϑ_i}, its colour x_t changes according to the stochastic flow $\phi_{\vartheta_i, x_{\tau_{k-1}}, t-\tau_{k-1}}$, i.e., $x_t = \phi_{\vartheta_i, x_{\tau_{k-1}}, t-\tau_{k-1}}$ de-

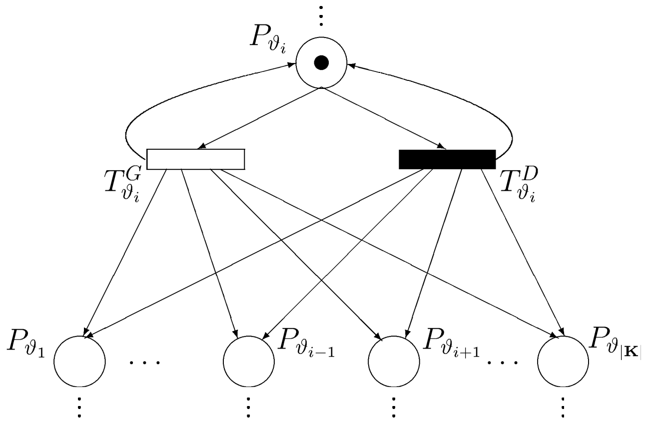


Fig. 2. Part of a Stochastically and Dynamically Coloured Petri Net representing a Generalised Stochastic Hybrid Process

fined on the complete probability space $(\Omega, \mathfrak{F}, \mathfrak{P}, \{\mathfrak{F}_t\})$. Transitions $T^G_{\vartheta_i}$ and $T^D_{\vartheta_i}$ are both pre-enabled and compete for this token which resides in their common input place P_{ϑ_i}. Transition $T^G_{\vartheta_i}$ models the boundary hitting generating a mode switch, while transition $T^D_{\vartheta_i}$ models the Poisson process generating a mode switch. For this, use is made of a random sample from the Hilbert cube. The transition that is enabled first, determines the kind of switch occurring. The time at which this happens is denoted by τ_k.

Step 2: With one, or more (has probability zero), of the transitions enabled at time τ_k, its firing measure is evaluated. For this, use is made of a random sample from the Hilbert cube. The firing measure is such, that if a sample ζ_k from transition measure $Q(\cdot; \vartheta_i, \phi_{\vartheta_i, x_{\tau_{k-1}}, \tau_k - \tau_{k-1}})$, would appear to be $\zeta_k = (\vartheta_j, x)$, then the enabled transition would produce one token with colour $x_{\tau_k} = x$ for place P_{ϑ_j}. The other places get no token.

After this, the above two steps are repeated in the same way from the new state on. The pathwise equivalence of the GSHP and SDCPN processes can be shown from the first stopping time to the next stopping time, and so on. From stopping time to stopping time both processes use the same independent realisations of the random variables U_1, U_2, ..., each having uniform $[0, 1]$ distribution, defined by $U_i(\omega) = \omega_i$ for elements $\omega = (\omega_1, \omega_2, \ldots)$ of the Hilbert cube $\Omega^H = \prod_{i=1}^{\infty} Y_i$, with Y_i a copy of $Y = [0, 1]$, to generate all random variables in both the GSHP process and the SDCPN process. Hence, from stopping time to stopping time, the GSHP and the associated SDCPN process have equivalent paths and equivalent stopping times. □

5 Stochastically and Dynamically Coloured Petri Nets into Generalised Stochastic Hybrid Processes

Under some conditions, each Stochastically and Dynamically Coloured Petri Net can be represented by a Generalised Stochastic Hybrid Process. In this section this is shown by providing an into-mapping from SDCPN into the set of GSHPs.

Theorem 2. *For each stochastic process generated by a Stochastically and Dynamically Coloured Petri Net* $(\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{C}, \mathcal{I}, \mathcal{V}, \mathcal{W}, \mathcal{G}, \mathcal{D}, \mathcal{F})$ *satisfying* R_0 *through* R_4 *there exists a unique probabilistically equivalent Generalised Stochastic Hybrid Process if the following conditions are satisfied:*

D_1 *There are no explosions, i.e. the time at which a token colour equals* $+\infty$ *or* $-\infty$ *approaches infinity whenever the time until the first guard transition enabling moment approaches infinity.*

D_2 *After a transition firing (or after a sequence of firings that occur at the same time instant) at least one place must contain a different number of tokens, or the colour of at least one token must have jumped*

D_3 *In a finite time interval, each transition is expected to fire a finite number of times, and for* $t \to \infty$ *the number of tokens remains finite.*

D_4 *The initial marking is such, that no immediate transition is initially enabled.*

Proof. For an arbitrary SDCPN that satisfies conditions D_1 – D_4, we first construct a GSHP that is probabilistically equivalent to the SDCPN process. As a preparatory step, the given SDCPN is enlarged as follows: for each guard transition and each place from which that guard transition may be enabled, copy the corresponding places and transitions, including guards and firing measures, and revise the firing measures of the input transitions to these places, such that the new firings ensure that the corresponding guard transitions may be reached from one side only. This step is illustrated with an example:

Example 1. In the picture on the left in Figure 3, transition T_1 (which may be of any type) may fire tokens to place P_1, while transition T_2 is a guard transition that uses these tokens as input. In this example, assume that $\mathcal{C}(P_1) = \mathbb{R}$ and that $\partial G_{T_2} = 3$. This means, transition T_2 is enabled if the colour of the token in place P_1 reaches value 3. This value may be reached from above or from below, depending on whether the initial colour of the token in P_1 is larger or smaller than 3, respectively.

In the picture on the right, place P_1 and transition T_2 have been copied. Transitions T_{2a} and T_{2b} get the same guard as T_2, but transition T_1' gets a new firing measure with respect to T_1: it is similar to the one of T_1, but it delivers a token to place P_{1a} if the colour of this new token is smaller than 3, and it delivers a token to place P_{1b} if its colour is larger than 3. This way, the

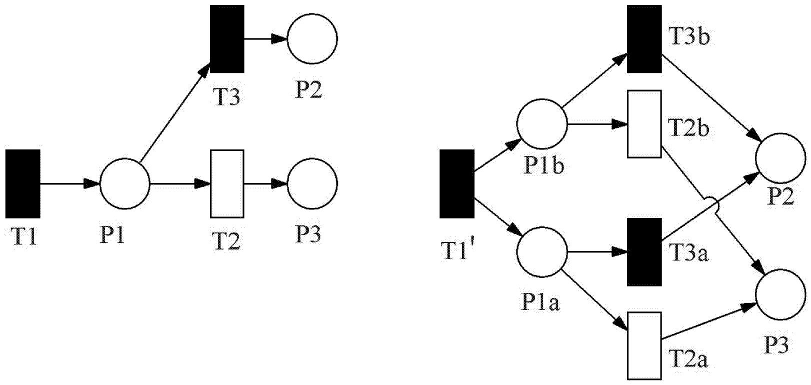


Fig. 3. Example transformation to model SDCPN enlargement

guard of transition T_{2a} is always reached from below, i.e., its input colours are smaller than 3. The guard of transition T_{2b} is always reached from above, i.e., its input colours are larger than 3. The second output transition T_3 of place P_1 also needs to be copied, but the output place of these copies can remain the same as before. (*End of Example*)

(Continuation of proof.) Let this enlarged SDCPN be described by the tuple ($\mathcal{P}$, $\mathcal{T}$, $\mathcal{A}$, $\mathcal{N}$, $\mathcal{S}$, $\mathcal{C}$, $\mathcal{I}$, $\mathcal{V}$, $\mathcal{W}$, $\mathcal{G}$, $\mathcal{D}$, $\mathcal{F}$) and satisfy the rules R_0 – R_4, and assume that the conditions D_1 – D_4 are satisfied. In order to represent this SDCPN by a GSHP, all GSHS elements $\mathbf{K}$, $d(\theta)$, x_0, θ_0, g_θ, g_θ^w, ∂E_θ, λ, Q and the GSHS conditions $C_1 - C_4$ are characterised in terms of this SDCPN:

K: The domain $\mathbf{K}$ for the mode process $\{\theta_t\}$ can be found from the reachability graph (RG) of the SDCPN graph. The nodes in the RG are vectors $V = (v_1, \ldots, v_{|\mathcal{P}|})$, where v_i equals the number of tokens in place P_i, $i = 1, \ldots, |\mathcal{P}|$, where these places are uniquely ordered. The RG is constructed from SDCPN components $\mathcal{P}$, $\mathcal{T}$, $\mathcal{A}$, $\mathcal{N}$ and $\mathcal{I}$. The first node V_0 is found from $\mathcal{I}$, which provides the numbers of tokens initially in each of the places[2]. From then on, the RG is constructed as follows: If it is possible to move in one jump from token distribution V_0 to, say, either one of distributions $V^1, \ldots, V^k$ unequal to V_0, then arrows are drawn from V_0 to (new) nodes $V^1, \ldots, V^k$. Each of $V^1, \ldots, V^k$ is treated in the same way. Each arrow is labelled by the (set of) transition(s) fired at the jump. If a node V^j can be directly reached from V^i by different (sets of) transitions firing, then multiple arrows are drawn from V^i to V^j, each labelled by another (set) of transition(s). Multiple arrows are also drawn if V^j can be directly reached from V^i by firing of one transition, but by different sets of tokens, for example in case this transition has multiple input tokens

[2] Notice that $\mathbf{K}$ has to be constructed for all $\mathcal{I}$ by following the proposed procedure such that is applies for each possible instantiation of the initial token distribution.

per incoming arc in its input places. In this case, the multiple arrows each get this transition as label.
The nodes in the resulting reachability graph, *exclusive* the nodes from which an immediate transition is enabled, form the discrete domain **K** of the GSHP. To emphasise these nodes from which an immediate transition is enabled in the RG picture, they are given in *italics*. Since the number of places in the SDCPN is finite and the number of tokens per place and the number of nodes in the RG are countable, **K** is a countable set, which satisfies the GSHS conditions.

Example 2. As an example, consider the SDCPN graph in Figure 4, which first is enlarged as explained above; the result is Figure 5. The enlarged graph initially has two tokens in place P_{1a} and one in P_3, and the unique ordering of places is $(P_{1a}, P_{1b}, P_2, P_3, P_4)$ such that $V_0 = (2, 0, 0, 1, 0)$. This vector forms the first node of the reachability graph.

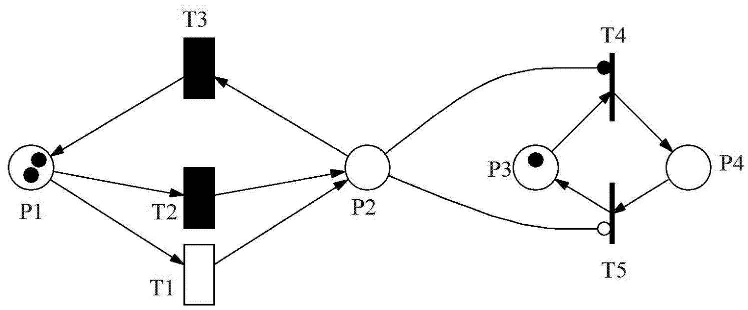


Fig. 4. Example SDCPN to explain reachability graph

Both T_{1a} and T_{2a} are pre-enabled. They both have two tokens per incoming arc in their input place, hence for both transitions, two vectors of input colours are evaluated in parallel. If T_{1a} becomes enabled for one of these input tokens, it removes the corresponding token from P_{1a} and produces a token for P_2 (we assume that all firing measures are such, that each transition will fire a token when enabled, i.e., $\mathcal{F}_T(0, \cdot; \cdot) = 0$), so the new token distribution is $(1, 0, 1, 1, 0)$. Therefore, in the reachability graph two arcs labelled by T_{1a} are drawn from $(2, 0, 0, 1, 0)$ to the new node $(1, 0, 1, 1, 0)$; this duplication of arcs characterises that T_{1a} has evaluated two vectors of input tokens in parallel. The same reasoning holds for transition T_{2a}: two arcs are drawn from $(2, 0, 0, 1, 0)$ to $(1, 0, 1, 1, 0)$. It may also happen that from $(2, 0, 0, 1, 0)$, the guard transition T_{1a} is enabled by its two input tokens at exactly the same time. Due to Rule R_1 it then fires these two tokens at exactly the same time, resulting in node $(0, 0, 2, 1, 0)$. Therefore, an additional arc labelled $T_{1a} + T_{1a}$ is drawn from $(2, 0, 0, 1, 0)$ to $(0, 0, 2, 1, 0)$. Unlike the case for T_{1a}, there is no arc

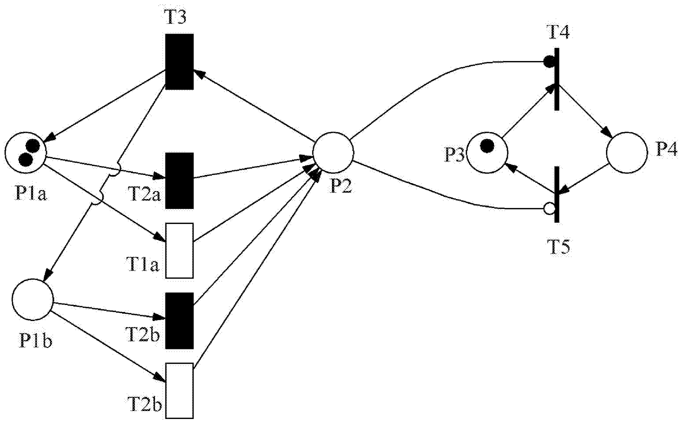


Fig. 5. Example enlarged SDCPN to explain reachability graph

drawn from $(2,0,0,1,0)$ labelled by $T_{2a} + T_{2a}$, since T_{2a} is a delay transition, hence the probability that it is enabled by both its input tokens at the same time is zero. Now consider node $(0,0,2,1,0)$. From this token distribution the immediate transition T_4 is enabled; its firing leads to $(1,0,1,0,1)$. Since node $(1,0,1,1,0)$ enables an immediate transition it is drawn in italics and is excluded from $\mathbf{K}$.

The resulting reachability graph for this example is given in Figure 6. So, for this example, $\mathbf{K} = \{(2,0,0,1,0),\ (0,0,2,0,1),\ (1,0,1,0,1),\ (0,1,1,0,1),\ (1,1,0,1,0),\ (0,2,0,1,0)\}$. (*End of Example*)

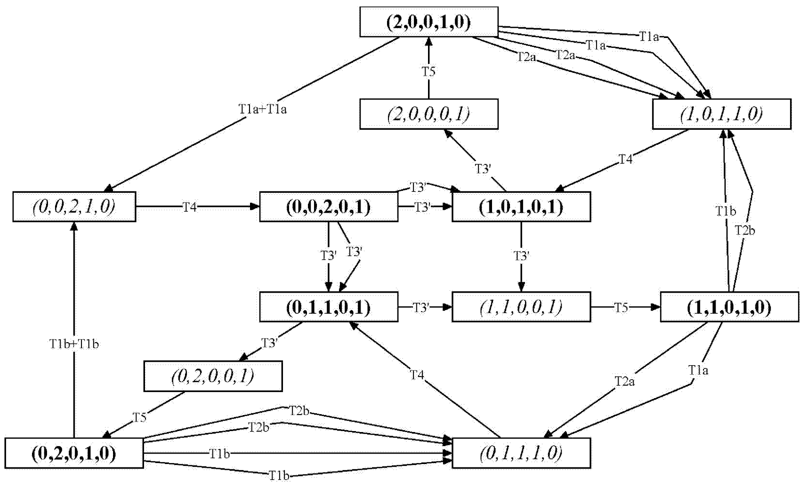


Fig. 6. Example reachability graph

(Continuation of proof.)

$d(\theta)$: The colour of a token in a place P is an element of $\mathcal{C}(P) = \mathbb{R}^{n(P)}$, therefore $d(\theta) = \sum_{i=1}^{|\mathcal{P}|} \theta_i \times n(P_i)$, with $\theta = (\theta_1, \ldots, \theta_{|\mathcal{P}|}) \in \mathbf{K}$, with $\{1, \ldots, |\mathcal{P}|\}$ referring to the unique ordering of places adopted for the SDCPN.

g_θ and g_θ^w: For $x = \text{Col}\{x^1, \ldots, x^{|\mathcal{P}|}\}$, with $x^i \in \mathbb{R}^{\theta_i \times n(P_i)}$, and with $\{1, \ldots,$ $|\mathcal{P}|\}$ referring to the unique ordering of places adopted for the SDCPN, g_θ is defined by $g_\theta(x) = \text{Col}\{g_\theta^1(x^1), \ldots, g_\theta^{|\mathcal{P}|}(x^{|\mathcal{P}|})\}$, where for $x^i = \text{Col}\{x^{i1},$ $\ldots, x^{i\theta_i}\}$, with $x^{ij} \in \mathbb{R}^{n(P_i)}$ for all $j \in \{1, \ldots, \theta_i\}$: $g_\theta^i(x^i) = \text{Col}\{\mathcal{V}_{P_i}(x^{i1}),$ $\ldots, \mathcal{V}_{P_i}(x^{i\theta_i})\}$. Here, $j \in \{1, \ldots, \theta_i\}$ refers to the unique ordering of tokens within their place defined for SDCPN (see Section 3). In a similar way, g_θ^w is defined by $g_\theta^w(x) = \text{Diag}\{g_\theta^{w,1}(x^1), \ldots, g_\theta^{w,|\mathcal{P}|}(x^{|\mathcal{P}|})\}$. Since, for all P_i, $\mathcal{V}_{P_i}$ and $\mathcal{W}_{P_i}$ satisfy conditions that ensure existence of a pathwise unique solution without explosion, this also applies to g_θ and g_θ^w.

∂E_θ: For each token distribution θ, the boundary ∂E_θ of subset E_θ is determined from the transition guards corresponding with the set of transitions in $\mathcal{T}_G$ that, under token distribution θ, are pre-enabled (this set is uniquely determined). Without loss of generality, suppose this set of transitions is $T_1, \ldots, T_m$ (note that this set may contain one transition multiple times, if multiple tokens are evaluated in parallel). Suppose $\{P^{i1}, \ldots, P^{ir_i}\}$ are the input places of T_i that are connected to T_i by means of ordinary or enabling arcs. Define $d_i = \sum_{j=1}^{r_i} n(P^{ij})$, then $\partial E_\theta = \partial G'_{T_1} \cup \ldots \cup \partial G'_{T_m}$, where $G'_{T_i} = [G_{T_i} \times \mathbb{R}^{d(\theta)-d_i}] \in \mathbb{R}^{d(\theta)}$. Here $[\cdot]$ denotes a special ordering of all vector elements: Vector elements corresponding with tokens in place P_a are ordered before vector elements corresponding with tokens in place P_b if $b > a$, according to the unique ordering of places adopted for the SDCPN; vector elements corresponding with tokens within one place are ordered according to the unique ordering of tokens within their place defined for SDCPN (see Section 3). If the set of pre-enabled guard transitions is empty, then $\partial E_\theta = \emptyset$.

λ: For each token distribution θ, the jump rate $\lambda(\theta, \cdot)$ is determined from the transition delays corresponding with the set of transitions in $\mathcal{T}_D$ that, under token distribution θ, are pre-enabled (this set is uniquely determined). Without loss of generality, suppose this set of transitions is $T_1, \ldots, T_m$. Then $\lambda(\theta, \cdot) = \sum_{i=1}^{m} \delta_{T_i}(\cdot)$. This equality is due to the fact that the combined arrival process of individual Poisson processes is again Poisson, with an arrival rate equal to the sum of all individual arrival rates. Since δ_T is integrable for all $T \in \mathcal{T}_D$, λ is also integrable. If the set of pre-enabled delay transitions is empty, then $\lambda(\theta, \cdot) = 0$.

Q: For each $\theta \in \mathbf{K}$, $x \in E_\theta$, $\theta' \in \mathbf{K}$ and $x' \in E_{\theta'}$, $Q(\theta', x'; \theta, x)$ is characterised by the reachability graph, the sets $\mathcal{D}$, $\mathcal{G}$ and $\mathcal{F}$ and the rules $R_0 - R_4$. The reachability graph is used to determine which transitions are pre-enabled in token distribution θ; the sets $\mathcal{D}$ and $\mathcal{G}$ and the rules $R_0 - R_4$ are used to determine which pre-enabled transitions will actually fire from state (θ, x); and finally, set $\mathcal{F}$ is used to determine the probability of (θ', x')

being the state after the jump, given state (θ, x) before the jump and the set of transitions that will fire in the jump. Because of its complexity, the characterisation of Q is given in the appendix, but an outline is given next:

Main challenge in the characterisation of Q is the following: In some situations one does not know for certain which transitions will fire in a jump, even if one knows the state (θ, x) before the jump and knows that a jump will occur from (θ, x) to (θ', x'). Hence, in these situations it is not known with certainty which firing measures one should combine in order to construct $Q(\theta', x'; \theta, x)$ from SDCPN elements. However, one does know the following:

- Given θ, one knows which transitions are pre-enabled; this can be read off the reachability graph (i.e. gather the labels of all arrows leaving node θ).
- Given that $\theta \in \mathbf{K}$, no immediate transitions are enabled in θ.
- The probability that a guard transition and a delay transition are enabled at exactly the same time is zero.
- The probability that two delay transitions are enabled at exactly the same time is zero.
- There is a possibility that two or more guard transitions are enabled at exactly the same time. It may even occur (due to rule R_1) that one single guard transition fires twice at the same time.

Hence, the steps to be followed to construct $Q(\theta', x'; \theta, x)$, for any (θ', x', θ, x) are:

1. Determine (using the reachability graph) which transitions are pre-enabled in θ.
2. Consider the guard transitions in this set of pre-enabled transitions and determine which of these are enabled. For a transition T, this is done by considering its vector of input colours (which is part of x) and checking whether this vector has entered the boundary ∂G_T. If the set of enabled guard transitions is not empty, then use rules $R_1 - R_4$ to find out which of these transitions will actually fire with which probability.
   If this set of enabled guard transitions is empty, then one pre-enabled delay transition must be enabled. Use $\mathcal{D}$ to determine for each pre-enabled delay transition the probability with which it will actually fire.
3. Determine which transition firings can actually lead to discrete process state θ' in one jump. This set can be found by identifying in the reachability graph all arrows directly from node θ to θ' and all directed paths from node θ to θ' that pass only nodes that enable immediate transitions (i.e. that pass only nodes in italics).
4. Finally, $Q(\theta', x'; \theta, x)$ is constructed from the firing measures, by conditioning on these arrows and paths from θ to θ'.

θ_0 and x_0: These can be constructed from $\mathcal{I}$, the SDCPN initial marking, which provides the places the tokens are initially in and the colours these tokens have. Hence, $\theta_0 = (v_{1,0}, \ldots, v_{|\mathcal{P}|,0})$, where $v_{i,0}$ denotes the initial number of tokens in place P_i, with the places ordered according to the unique ordering adopted for SDCPN, and $x_0 \in \mathbb{R}^{d(\theta_0)}$ is a vector containing the colours of these tokens. Within a place the colours of the tokens are ordered according to the specification in $\mathcal{I}$. With this, and due to condition D_4 (which prevents different token distributions to be applicable at the initial time), the constructed θ_0 and x_0 are uniquely defined.

C_1: This condition (no explosions) follows from assumption D_1.

C_2: This condition (λ is integrable) follows from the fact that δ_T is integrable for all $T \in \mathcal{T}_D$.

C_3: This condition (Q measurable and $Q(\{\xi\};\xi) = 0$) follows from the assumption that $\mathcal{F}$ is continuous and from assumption D_2.

C_4: This condition ($\mathbb{E}N_t < \infty$) follows from assumption D_3.

This shows that for any SDCPN satisfying conditions D_1 – D_4, we are able to construct unique GSHS elements, and thus a unique GSHS.

Finally, we show that the GSHP process $\{\theta_t, x_t\}$ is probabilistically equivalent to the process generated by the SDCPN:

With the mapping from SDCPN elements into GSHS elements, it is easily shown that the GSHP process $\{\theta_t, x_t\}$ is probabilistically equivalent to the process generated by the SDCPN characterised in Section 3: at each time t the process $\{\theta_t\}$ is probabilistically equivalent to the process $(v_{1,t}, \ldots, v_{|\mathcal{P}|,t})$ and the process $\{x_t\}$ is probabilistically equivalent to the process associated with the vector of token colours. This is shown by observing that the initial GSHP state (θ_0, x_0) is probabilistically equivalent to the initial SDCPN state through the mapping constructed above. Moreover, also by the unique mapping of SDCPN elements into GSHS elements, at each time instant after the initial time, the GSHP state is probabilistically equivalent to the SDCPN state: At times t when no jump occurs, the GSHP process evolves according to g_θ and g_θ^w and the SDCPN process evolves according to $\mathcal{V}$ and $\mathcal{W}$. Through the mapping between g_θ and $\mathcal{V}$ and between g_θ^w and $\mathcal{W}$ developed above, these evolutions provide probabilistically equivalent processes. At times when a jump occurs, the GSHP process makes a jump generated by Q, while the SDCPN process makes a jump generated by $\mathcal{F}$. Through the mapping between Q and $\mathcal{F}$ developed above, these jumps provide probabilistically equivalent processes.

6 Example SDCPN and Mapping to GSHP

This section gives a simple example SDCPN model and its mapping to GSHP of the evolution of an aircraft. First, Subsection 6.1 explains how a SDCPN that models a complex operation is generally constructed in three steps. In order to illustrate these steps, Subsection 6.2 presents a simple example of the evolution of one aircraft. Subsection 6.3 gives a SDCPN that models this aircraft evolution and Subsection 6.4 explains the mapping of this SDCPN example in a GSHP.

6.1 SDCPN Construction and Verification Process

A SDCPN modelling a particular operation can be constructed, for example, by first identifying the discrete state space, represented by the places, the transitions and arcs, and next adding the continuous-time-based elements one by one, similar as what one would expect when modelling a GSHP for such operation. However, in case of a very complex operation, with many entities that interact such as occur in air traffic, it is generally more desirable and constructive to do the SDCPN modelling in several iterations, for example in a four-phased approach:

1. In the first phase, each operation entity or agent (for example, a pilot, a navigation system, an aircraft) is modelled separately by one local DCPN (i.e. no Brownian motion components $\mathcal{W}$). Each such entity model is named a Local Petri Net (LPN).
2. In the second phase, the interactions between these entities are modelled, connecting the LPNs, such that these interactions do not change the number of tokens per LPN.
3. In the third phase the Brownian motion components $\mathcal{W}$ are added to the LPNs.
4. In the fourth phase, one verifies whether the conditions D_1 – D_4 under which a mapping to GSHP is guaranteed to exist have been fulfilled. Because of the modularity and fixed number of tokens per LPN, these conditions can easily be verified per LPN, and subsequently per interaction between LPNs.

The additional advantage of this phased approach is that the total SDCPN can be verified simultaneously by multiple domain experts. For example, a Local Petri Net model for a navigation system can be verified by a navigational system expert; a Local Petri Net model for a pilot can be verified by a human factors expert; interactions can be verified by a pilot.

6.2 Aircraft Evolution Example

This subsection presents a simple aircraft evolution example. The next subsections present a SDCPN model and a mapping to GSHP for this example.

Assume the deviation of this aircraft from its intended path depends on the operationality of two of its aircraft systems: the engine system, and the navigation system. Each of these aircraft systems can be in one of two modes: *Working* (functioning properly) or *Not working* (operating in some failure mode). Both systems switch between their modes independently and on exponentially distributed times, with rates δ_3 (engine repaired), δ_4 (engine fails), δ_5 (navigation repaired) and δ_6 (navigation fails), respectively. The operationality of these systems has the following effect on the aircraft path: if both systems are *Working*, the aircraft evolves in *Nominal* mode and the rate of change of the position and velocity of the aircraft is determined by $(\mathcal{V}_1, \mathcal{W}_1)$ (i.e. if z_t is a vector containing this position and velocity then $dz_t = \mathcal{V}_1(z_t)dt + \mathcal{W}_1 dw_t$). If either one, or both, of the systems is *Not working*, the aircraft evolves in *Non-nominal* mode and the position and velocity of the aircraft is determined by $(\mathcal{V}_2, \mathcal{W}_2)$. The factors $\mathcal{W}_1$ and $\mathcal{W}_2$ are determined by wind fluctuations. Initially, the aircraft has a particular position x_0 and velocity v_0, while both its systems are *Working*. The evaluation of this process may be stopped when the aircraft position has *Landed*, i.e. its vertical position and velocity is equal to zero. Once landed, the aircraft is assumed not to depart anymore, hence the rate of change of its position and velocity equals zero.

This simple aircraft evolution example illustrates the kind of difficulty encountered when one wants to model a realistic problem directly as a GSHP. Mathematically one would define three discrete valued processes $\{\kappa_t^1\}$, $\{\kappa_t^2\}$, $\{\kappa_t^3\}$, and an $\mathbb{R}^6$-valued process $\{x_t\}$:

- $\{\kappa_t^1\}$ represents the aircraft evolution mode assuming values in {*Nominal*, *Non-nominal*, *Landed*};
- $\{\kappa_t^2\}$ represents the navigation mode assuming values in {*Working*, *Not-working*};
- $\{\kappa_t^3\}$ represents the engine mode assuming values in {*Working*, *Not-working*};
- $\{x_t\}$ represents the 3D position and 3D velocity of the aircraft

Unfortunately, the process $\{\kappa_t, x_t\}$, with $\kappa_t = \mathrm{Col}\{\kappa_t^1, \kappa_t^2, \kappa_t^3\}$, is not a GSHP, since some κ_t combinations lead to immediate jumps, which is not allowed for GSHP.

6.3 SDCPN Model for the Aircraft Evolution Example

This subsection gives a SDCPN instantiation that models the aircraft evolution example of the previous subsection. In order to illustrate the three-phased approach of subsection 6.1, we first give the Local Petri Net graphs that have been identified in the first phase of the modelling. The entities identified are: Aircraft evolution, Navigation system, and Engine system. This gives us three Local Petri Nets. The resulting graphs are given in Figure 7.

The interactions between the Engine and Navigation Local Petri Net and the Evolution Local Petri Net are modelled by coupling the Local Petri Nets by additional arcs (and, if necessary, additional places or transitions). Here, removal of a token from one Local Petri Net by a transition of another Local Petri Net is prevented by using enabling arcs instead of ordinary arcs for the interactions. The resulting graph is presented in Figure 8. Notice that transition T_1 has to be replaced by two transitions T_{1a} and T_{1b} in order to

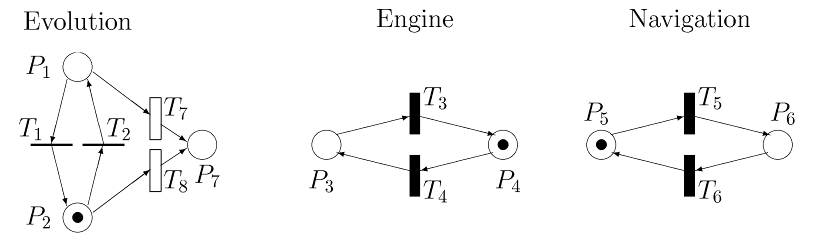


Fig. 7. Local Petri Nets for the aircraft operations example. Place P_1 models Evolution Nominal, P_2 models Evolution Non-nominal, P_3 models Engine system Not working, P_4 models Engine system Working, P_5 models Navigation system Not working, P_6 models Navigation system Working. P_7 models aircraft has landed

allow both the engine and the navigation LPNs to influence transition T_1 separately from each other.

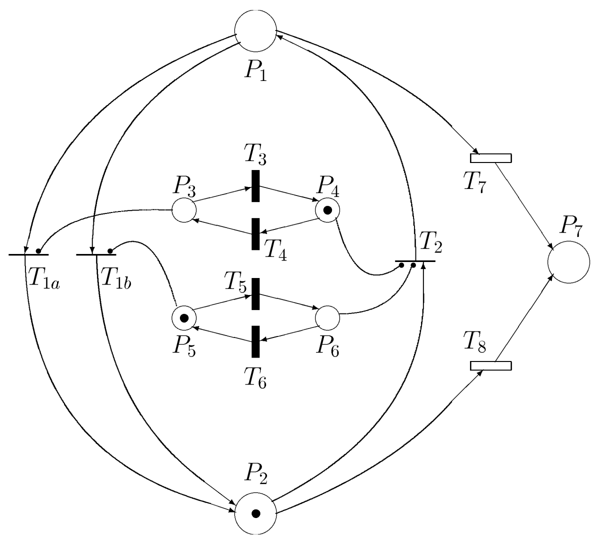


Fig. 8. Local Petri Nets integrated into one Petri Net

The graph above completely defines SDCPN elements $\mathcal{P}$, $\mathcal{T}$, $\mathcal{A}$ and $\mathcal{N}$, where $\mathcal{T}_G = \{T_7, T_8\}$, $\mathcal{T}_D = \{T_3, T_4, T_5, T_6\}$ and $\mathcal{T}_I = \{T_{1a}, T_{1b}, T_2\}$. The other SDCPN elements are specified below.

$\mathcal{S}$: Two colour types are defined; $\mathcal{S} = \{\mathbb{R}^0, \mathbb{R}^6\}$.

$\mathcal{C}$: $\mathcal{C}(P_1) = \mathcal{C}(P_2) = \mathcal{C}(P_7) = \mathbb{R}^6$, hence $n(P_1) = n(P_2) = n(P_7) = 6$. The first three colour components model the longitudinal, lateral and vertical position of the aircraft, the last three components model the corresponding velocities. For places P_3 through P_6, $\mathcal{C}(P_i) = \mathbb{R}^0 = \emptyset$ hence $n(P_i) = 0$.

$\mathcal{I}$: Place P_1 initially has a token with colour $z_0 = (x_0, v_0)'$, with $x_0 \in \mathbb{R}^2 \times (0, \infty)$ and $v_0 \in \mathbb{R}^3 \backslash \text{Col}\{0, 0, 0\}$. Places P_4 and P_6 initially each have a token without colour.

$\mathcal{V}$ and

$\mathcal{W}$: The token colour functions for places P_1, P_2 and P_7 are determined by $(\mathcal{V}_1, \mathcal{W}_1)$, $(\mathcal{V}_2, \mathcal{W}_2)$, and $(\mathcal{V}_7, \mathcal{W}_7)$, respectively, where $(\mathcal{V}_7, \mathcal{W}_7) = (0, 0)$. For places P_3 – P_6 there is no token colour function.

$\mathcal{G}$: Transitions T_7 and T_8 have a guard that is defined by $\partial G_{T_7} = \partial G_{T_8} = \mathbb{R}^2 \times \{0\} \times \mathbb{R}^2 \times \{0\}$.

$\mathcal{D}$: The enabling rates for transitions T_3, T_4, T_5 and T_6 are $\delta_{T_3}(\cdot) = \delta_3$, $\delta_{T_4}(\cdot) = \delta_4$, $\delta_{T_5}(\cdot) = \delta_5$ and $\delta_{T_6}(\cdot) = \delta_6$, respectively.

$\mathcal{F}$: Each transition has a unique output place, to which it fires to their output place a token with a colour (if applicable) equal to the colour of the token removed, i.e. for all T, $\mathcal{F}_T(1, \cdot; \cdot) = 1$.

6.4 Mapping to GSHP

In this subsection, the SDCPN aircraft evolution example is mapped to a GSHP, following the construction in the proof of Theorem 2. Because the boundaries of the guard transitions T_7 and T_8 (i.e. $\partial G_{T_7} = \partial G_{T_8} = \mathbb{R}^2 \times \{0\} \times \mathbb{R}^2 \times \{0\}$) are always reached from one side only, there is no need to first enlarge the SDCPN for these guard transitions (see Section 5).

The SDCPN of Figure 8 has seven places hence the reachability graph has elements that are vectors of length 7. Since there is always one token in the set of places $\{P_1, P_2, P_7\}$, one token in $\{P_3, P_4\}$ and one token in $\{P_5, P_6\}$, the reachability graph has $3 \times 2 \times 2 = 12$ nodes, see Figure 9. However, four nodes are excluded from $\mathbf{K}$: nodes $(1, 0, 1, 0, 0, 1, 0)$, $(0, 1, 0, 1, 0, 1, 0)$ and $(1, 0, 0, 1, 1, 0, 0)$ enable immediate transitions, and node $(1, 0, 1, 0, 1, 0, 0)$ cannot be reached since it requires the enabling of a delay transition that is competing with an immediate transition, while due to SDCPN rule R_0, an immediate transition always gets priority. Therefore, $\mathbf{K}$ consists of the remaining 8 nodes $\{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8\}$, which are specified in Table 1.

Table 1. Discrete modes in $\mathbf{K}$

Node	Engine	Navigation	Evolution
$m_1 = (1, 0, 0, 1, 0, 1, 0)$	*Working*	*Working*	*Nominal*
$m_2 = (0, 1, 1, 0, 0, 1, 0)$	*Not working*	*Working*	*Non-nominal*
$m_3 = (0, 1, 1, 0, 1, 0, 0)$	*Not working*	*Not working*	*Non-nominal*
$m_4 = (0, 1, 0, 1, 1, 0, 0)$	*Working*	*Not working*	*Non-nominal*
$m_5 = (0, 0, 0, 1, 0, 1, 1)$	*Working*	*Working*	*Landed*
$m_6 = (0, 0, 1, 0, 0, 1, 1)$	*Not working*	*Working*	*Landed*
$m_7 = (0, 0, 1, 0, 1, 0, 1)$	*Not working*	*Not working*	*Landed*
$m_8 = (0, 0, 0, 1, 1, 0, 1)$	*Working*	*Not working*	*Landed*

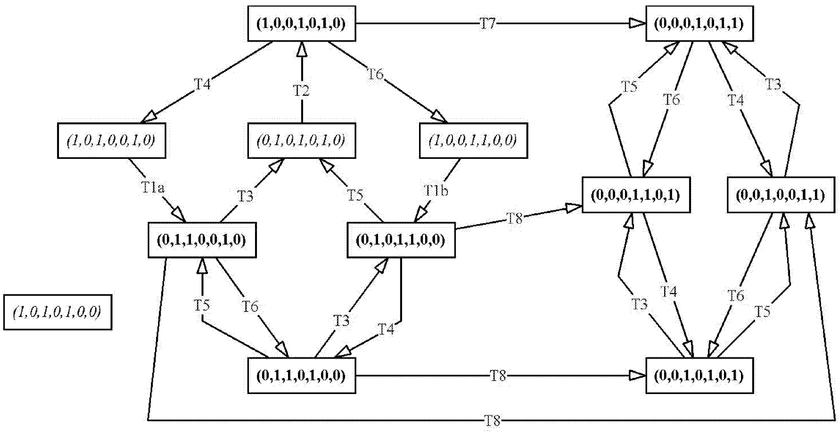


Fig. 9. Reachability graph for the SDCPN of Figure 8

Following Section 5, for each $\theta = (\theta_1, \ldots, \theta_7) \in \mathbf{K}$, the value of $d(\theta)$ equals $d(\theta) = \sum_{i=1}^{|\mathcal{P}|} \theta_i \times n(P_i)$. Since there is always one token in the set of places $\{P_1, P_2, P_7\}$, hence $\theta_1 + \theta_2 + \theta_7 = 1$, and since $n(P_1) = n(P_2) = n(P_7) = 6$ and $n(P_3) = n(P_4) = n(P_5) = n(P_6) = 0$, we find for all θ that $d(\theta) = 6$.

Since initially there is a token in places P_1, P_4 and P_6, the initial mode θ_0 equals $\theta_0 = m_1 = (1, 0, 0, 1, 0, 1, 0)$. The GSHP initial continuous state value equals the vector containing the initial colours of all initial tokens. Since the initial colour of the token in Place P_1 equals z_0, and the tokens in places P_4 and P_6 have no colour, the GSHP initial continuous state value equals z_0.

Following Section 5, with $\theta = (\theta_1, \ldots, \theta_7) \in \mathbf{K}$, for $x = \text{Col}\{x^1, \ldots, x^7\}$, with $x^i \in \mathbb{R}^{\theta_i \times n(P_i)}$, the function g_θ is defined by $g_\theta(x) = \text{Col}\{g_\theta^1(x^1), \ldots, g_\theta^7(x^7)\}$, where for $x^i = \text{Col}\{x^{i1}, \ldots, x^{i\theta_i}\}$, with $x^{ij} \in \mathbb{R}^{n(P_i)}$ for all $j \in \{1, \ldots, \theta_i\}$: $g_\theta^i(x^i)$ satisfies $g_\theta^i(x^i) = \text{Col}\{\mathcal{V}_{P_i}(x^{i1}), \ldots, \mathcal{V}_{P_i}(x^{i\theta_i})\}$. Since there is at most one token in each place, θ_i is either zero or one, hence either $x^i = \emptyset$ or $x^i = x^{i1}$. Since there is no token colour function for places $\{P_3, P_4, P_5, P_6\}$ and there is only one token in $\{P_1, P_2, P_7\}$, $g_\theta(x) = \mathcal{V}_1$ for $\theta = m_1$, $g_\theta(x) = \mathcal{V}_2$ for $\theta \in \{m_2, m_3, m_4\}$, and $g_\theta(x) = 0$ otherwise. In a similar way, $g_\theta^w(x) = \mathcal{W}_1$ for $\theta = m_1$, $g_\theta^w(x) = \mathcal{W}_2$ for $\theta \in \{m_2, m_3, m_4\}$, and $g_\theta^w(x) = 0$ otherwise, see Table 2.

The boundary ∂E_θ is determined from the transitions guards that, under token distribution θ, are enabled. This yields: for $\theta = m_1$, $\partial E_\theta = \partial G_{T_7} = \mathbb{R}^2 \times \{0\} \times \mathbb{R}^2 \times \{0\}$; for $\theta \in \{m_2, m_3, m_4\}$, $E_\theta = \partial G_{T_8} = \mathbb{R}^2 \times \{0\} \times \mathbb{R}^2 \times \{0\}$; for $\theta \in \{m_5, m_6, m_7, m_8\}$, $\partial E_\theta = \emptyset$.

The jump rate $\lambda(\theta, \cdot)$ is determined from the enabling rates corresponding with the set of delay transitions in $\mathcal{T}_D$ that, under token distribution θ, are pre-enabled. At each time, always two delay transitions are pre-enabled: either

T_3 or T_4 and either T_5 or T_6. Hence $\lambda(\theta, \cdot) = \sum_{i=j,k} \delta_{T_i}(\cdot)$ if T_j and T_k are pre-enabled. See Table 2 for the resulting λ's.

The probability measure Q is determined by the reachability graph, the sets $\mathcal{D}$, $\mathcal{G}$ and $\mathcal{F}$ and the rules $R_0 - R_4$. In Table 3, $Q(\zeta; \xi) = p$ denotes that if ξ is the value of the GSHP before the hybrid jump, then, with probability p, ζ is the value of the GSHP immediately after the jump.

Table 2. Example GSHS components $g_\theta(\cdot)$, $g^w_\theta(\cdot)$ and λ as a function of θ

θ	$g_\theta(\cdot)$	$g^w_\theta(\cdot)$	λ
m_1	$\mathcal{V}_1(\cdot)$	$\mathcal{W}_1(\cdot)$	$\delta_4 + \delta_6$
m_2	$\mathcal{V}_2(\cdot)$	$\mathcal{W}_2(\cdot)$	$\delta_3 + \delta_6$
m_3	$\mathcal{V}_2(\cdot)$	$\mathcal{W}_2(\cdot)$	$\delta_3 + \delta_5$
m_4	$\mathcal{V}_2(\cdot)$	$\mathcal{W}_2(\cdot)$	$\delta_4 + \delta_5$
m_5	0	0	$\delta_4 + \delta_6$
m_6	0	0	$\delta_3 + \delta_6$
m_7	0	0	$\delta_3 + \delta_5$
m_8	0	0	$\delta_4 + \delta_5$

Table 3. Example GSHS component Q

For $z \notin \partial E_{m_1}$:	$Q(m_2, z; m_1, z) = \frac{\delta_4}{\delta_4+\delta_6}$, $Q(m_4, z; m_1, z) = \frac{\delta_6}{\delta_4+\delta_6}$
For $z \in \partial E_{m_1}$:	$Q(m_5, z; m_1, z) = 1$
For $z \notin \partial E_{m_2}$:	$Q(m_3, z; m_2, z) = \frac{\delta_6}{\delta_3+\delta_6}$, $Q(m_1, z; m_2, z) = \frac{\delta_3}{\delta_3+\delta_6}$
For $z \in \partial E_{m_2}$:	$Q(m_6, z; m_2, z) = 1$
For $z \notin \partial E_{m_3}$:	$Q(m_4, z; m_3, z) = \frac{\delta_3}{\delta_3+\delta_5}$, $Q(m_2, z; m_3, z) = \frac{\delta_5}{\delta_3+\delta_5}$
For $z \in \partial E_{m_3}$:	$Q(m_7, z; m_3, z) = 1$
For $z \notin \partial E_{m_4}$:	$Q(m_3, z; m_4, z) = \frac{\delta_4}{\delta_4+\delta_5}$, $Q(m_1, z; m_4, z) = \frac{\delta_5}{\delta_4+\delta_5}$
For $z \in \partial E_{m_4}$:	$Q(m_8, z; m_4, z) = 1$
For all z:	$Q(m_6, z; m_5, z) = \frac{\delta_4}{\delta_4+\delta_6}$, $Q(m_8, z; m_5, z) = \frac{\delta_6}{\delta_4+\delta_6}$
For all z:	$Q(m_7, z; m_6, z) = \frac{\delta_6}{\delta_3+\delta_6}$, $Q(m_5, z; m_6, z) = \frac{\delta_3}{\delta_3+\delta_6}$
For all z:	$Q(m_8, z; m_7, z) = \frac{\delta_3}{\delta_3+\delta_5}$, $Q(m_6, z; m_7, z) = \frac{\delta_5}{\delta_3+\delta_5}$
For all z:	$Q(m_7, z; m_8, z) = \frac{\delta_4}{\delta_4+\delta_5}$, $Q(m_5, z; m_8, z) = \frac{\delta_5}{\delta_4+\delta_5}$

From a mathematical perspective, the GSHP model has clear advantages. However, the GSHP model does not show the structure of the SDCPN. Because of this, the SDCPN model of Subsection 6.3 is simpler to comprehend and to verify against the aircraft evolution example description of Subsection 6.2. These complementary advantages from both perspectives tend to increase with the complexity of the operation considered.

7 Conclusions

Generalised Stochastic Hybrid Processes (GSHPs) can be used to describe virtually all complex continuous-time stochastic processes. However, for complex practical problems it is often difficult to develop a GSHP model, and have it verified both by mathematical and by multiple operational domain experts. This paper has introduced a novel Petri Net, which is named Stochastically and Dynamically Coloured Petri Net (SDCPN) and has shown that under some mild conditions, any SDCPN generated process can be mapped into a probabilistically equivalent GSHP. Moreover, it is shown that any GSHP with a finite discrete state domain can be mapped into a pathwise equivalent process which is generated by a executing a GSHS. A consequence of both results is that there exist into-mappings between GSHPs and SDCPN processes. The development of a SDCPN model for complex practical problems has similar specification advantages as basic Petri Nets have over automata [4].

The key result of this paper is that this is the first time that proof of the existence of into-mappings between GSHPs and Petri Nets has been established. This significantly extends the modelling power hierarchy of [14],[15] in terms of Petri Nets and Markov processes, see Figure 10.

To the authors' best knowledge, SDCPN is the only hybrid Petri Net that incorporates Brownian motion. Moreover, SDCPN and DCPN are the only hybrid Petri Nets for which into-mappings with hybrid state Markov processes are known. Due to the existence of these into-mappings, GSHP theoretical results like stochastic analysis, stability and control theory, also apply to SDCPN stochastic processes. The mapping of SDCPN into GSHP implies that any specific SDCPN stochastic process can be analysed as if it is a GSHP, often without the need to first apply the transformation into a GSHP as we did for the aircraft evolution example in Section 6. Because of this, for accident risk modelling in air traffic management, in [2] SDCPNs are adopted for their specification power and for their GSHP inherited stochastic analysis power.

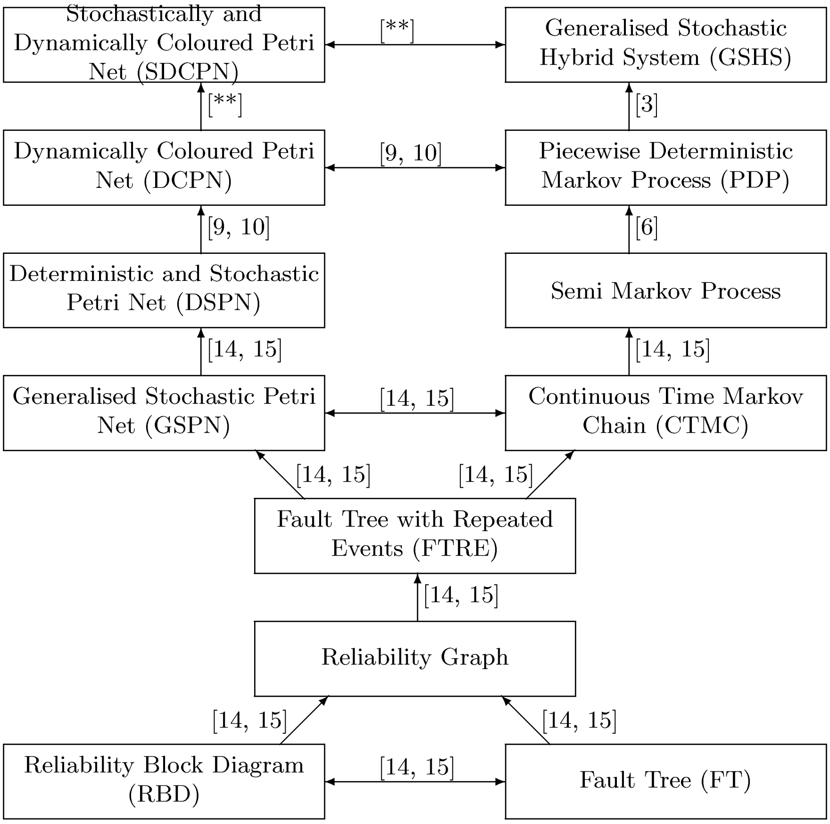


Fig. 10. Power hierarchy among various model types established by [6], [9], [10], [14], [15], [3] and the current paper (denoted by [**]). An arrow from a model to another model indicates that the second model has more modelling power than the first model

References

1. J. Le Bail, H. Alla, and R. David. Hybrid Petri nets. *Eropean Control Conference, Grenoble, France*, pages 1472–1477, 1991.
2. H.A.P. Blom, G.J. Bakker, P.J.G. Blanker, J. Daams, M.H.C. Everdij, and M.B. Klompstra. Accident risk assessment for advanced ATM. *2^{nd} USA/Europe Air Traffic Management R&D Seminar, Orlando*, 1998. Also in: *Air Transportation Systems Engineering*, AIAA, Eds. G.L. Donohue, A.G. Zellweger, AIAA, pp. 463-480 (2001).
3. M.L. Bujorianu, J. Lygeros, W. Glover, and G. Pola. A stochastic hybrid system modelling framework. Technical report, University of Cambridge and University of L'Aquila, May 2003. Hybridge report D1.2, Also a chapter in this book.
4. C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.

5. R. David and H. Alla. Petri nets for the modeling of dynamic systems - a survey. *Automatica*, 30(2):175–202, 1994.
6. M.H.A. Davis. Piecewise Deterministic Markov Processes: a general class of non-diffusion stochastic models. *Journal Royal Statistical Soc. (B)*, 46:353–388, 1984.
7. M.H.A. Davis. *Markov models and optimization.* Chapman and Hall, 1993.
8. I. Demongodin and N.T. Koussoulas. Differential Petri nets: Representing continuous systems in a discrete-event world. *IEEE Transactions on Automatic Control*, 43(4), 1998.
9. M.H.C. Everdij and H.A.P. Blom. Petri nets and hybrid state Markov processes in a power-hierarchy of dependability models. *Proc. IFAC Conference on Analysis and Design of Hybrid System (ADHS03), Saint-Malo, Brittany, France*, pages 355–360, June 2003.
10. M.H.C. Everdij and H.A.P. Blom. Piecewise Deterministic Markov Processes represented by Dynamically Coloured Petri Nets. *Stochastics*, 77(1):1–29, February 2005.
11. M.H.C. Everdij, H.A.P. Blom, and M.B. Klompstra. Dynamically Coloured Petri Nets for air traffic management safety purposes. *Proc. 8th IFAC Symposium on Transportation Systems, Chania, Greece*, pages 184–189, 1997.
12. A. Giua and E. Usai. High-level hybrid etri nets: a definition. *Proceedings 35th Conference on Decision and Control, Kobe, Japan*, pages 148–150, 1996.
13. K. Jensen. *Coloured Petri Nets: Basic concepts, analysis methods and practical use*, volume 1. Springer-Verlag, 1992.
14. M. Malhotra and K.S. Trivedi. Power-hierarchy of dependability-model types. *IEEE Transactions on Reliability*, R-43(3):493–502, 1994.
15. J.K. Muppala, R.M. Fricks, and K.S. Trivedi. Techniques for system dependability evaluation. In W. Grasman, editor, *Computational probability*, pages 445–480. Kluwer Academix Publishers, The Netherlands, 2000.
16. K.S. Trivedi and V.G. Kulkarni. FSPNs: Fluid stochastic Petri nets. In M. Ajmone Marsan, editor, *Proceedings 14th International Conference on Applications and theory of Petri Nets*, volume 691 of *Lecture notes in Computer Science*, pages 24–31. Springer Verlag, Heidelberg, 1993.
17. Y.Y. Yang, D.A. Linkens, and S.P. Banks. Modelling of hybrid systems based on extended coloured Petri nets. In P. Antsaklis et al, editor, *Hybrid Systems II*, pages 509–528. Springer, 1995.

A Characterisation of Q in Terms of SDCPN Elements

In this appendix, Q is characterised in terms of SDCPN, as part of the characterisation in Appendix C of GSHP in terms of SDCPN.

For each $\theta \in \mathbf{K}$, $x \in E_\theta$, $\theta' \in \mathbf{K}$ and $A \subset E_{\theta'}$, the value of $Q(\theta', A; \theta, x)$ is a measure for the probability that if a jump occurs, and if the value of the GSHP just prior to the jump is (θ, x), then the value of the GSHP just after the jump is in (θ', A). Measure $Q(\theta', A; \theta, x)$ is characterised in terms of the SDCPN by the reachability graph (RG) (see Appendix C), elements $\mathcal{D}$, $\mathcal{G}$ and Rules $R_0 - R_4$ and the set $\mathcal{F}$, as below. This is done in four steps:

1. Determine which transitions are pre-enabled in (θ, x).

2. Determine for each pre-enabled transition the probability with which it is enabled in (θ, x).
3. Determine for each pre-enabled transition whether its firing can possibly lead to discrete state θ'.
4. Use the results of the previous two steps and the set of firing functions to characterise Q.

Step 1: Determine which transitions are pre-enabled in $(\boldsymbol{\theta}, \boldsymbol{x})$.

Consider all arrows in the RG leaving node θ. These arrows are labelled by names of transitions which are pre-enabled in θ, for example T_1 (if T_1 is pre-enabled in θ), T_1+T_2 (if T_1 and T_2 are both pre-enabled and there is a non-zero probability that they fire at exactly the same time), etc. Therefore the arrows leaving θ may be characterised by these labels. Denote the multi-set of arrows, characterised by these labels, by $\mathcal{B}_\theta$. This set is a multi-set since there may exist several arrows with the same label (e.g. if one transition is pre-enabled by different sets of input tokens). We use notation $B \in \mathcal{B}_\theta$ for an element B of $\mathcal{B}_\theta$ (e.g. $B = T_1$ represents an arrow with T_1 as label), and notation $T \in B$ for a transition T in label B (e.g. as in $B = T + T_1$).

Step 2: Determine for each pre-enabled transition the probability with which it is enabled in $(\boldsymbol{\theta}, \boldsymbol{x})$.

Given that a jump occurs in (θ, x), the set of transitions that will actually fire in (θ, x) is not empty, and is given by one of the labels in $\mathcal{B}_\theta$. In the following, we determine, for all $B \in \mathcal{B}_\theta$, the probability $p_B(\theta, x)$ that all transitions in label B will fire.

- Denote the vector of input colours of transition T in a particular label by c_T^x. For a transition in a label this vector is unique since we consider transitions with multiple vectors of input colours separately in the multi-set $\mathcal{B}_\theta$.
- Consider the multi-set $\mathcal{B}_\theta^G = \{B \in \mathcal{B}_\theta | \forall T \in B : T \in \mathcal{T}_G \text{ and } c_T^x \in \partial G_T\}$.
- If $\mathcal{B}_\theta^G \neq \emptyset$ then this set contains all transitions that are enabled in (θ, x). Rules $R_1 - R_4$ are used (R_0 is not applicable) to determine for each $B \in \mathcal{B}_\theta^G$ the probability with which the transitions in label B will actually fire:
  - Rules R_1 and R_3 are used as follows: if B is such that there exists $B' \in \mathcal{B}_\theta^G$ such that the transitions in B form a real subset of the set of transitions in B', then $p_B(\theta, x) = 0$. The set of thus eliminated labels B is denoted by $\mathcal{B}_\theta^{R_{1,3}}$.
  - Rules R_2 and R_4 are used as follows: If the multi-set $\mathcal{B}_\theta^G - \mathcal{B}_\theta^{R_{1,3}}$ contains m elements, then each of these labels gets a probability $p_B(\theta, x) = 1/m$.

- If $\mathcal{B}_\theta^G = \emptyset$ then only Delay transitions can be enabled in (θ, x). Consider the multi-set $\mathcal{B}_\theta^D = \{B \in \mathcal{B}_\theta | \forall T \in B : T \in \mathcal{T}_D\}$. Each $B \in \mathcal{B}_\theta^D$ consists of one delay transition, with $p_B(\theta, x) = \frac{\delta_B(c_B^x)}{\sum\limits_{T \in \mathcal{B}_\theta^D} \delta_T(c_T^x)}$.

Step 3: Determine for each pre-enabled transition whether its firing can possibly lead to discrete state θ'.

In the RG, consider nodes θ and θ' and delete all other nodes that are elements of $\mathbf{K}$, including the arrows attached to them. Also, delete all nodes and arrows that are not part of a directed path from θ to θ'. The residue is named $\mathrm{RG}_{\theta\theta'}$. Then, if θ and θ' are not connected in $\mathrm{RG}_{\theta\theta'}$ by at least one path, a jump from (θ, x) to a state in (θ', A) is not possible.

Step 4: Use the results of the previous two steps and the set of firing functions to characterise Q.

From the previous step we have

- $Q(\theta', A; \theta, x) = 0$ if θ and θ' are not connected in $\mathrm{RG}_{\theta\theta'}$ by at least one path.

If θ and θ' are connected then in $\mathrm{RG}_{\theta\theta'}$ one or more paths from θ to θ' can be identified. Each such path may consist of only one arrow, or of sequences of directed arrows that pass nodes that enable immediate transitions. All arrows are labelled by names of transitions, therefore the paths between θ and θ' may be characterised by the labels on these arrows, i.e. by the transitions that consecutively fire in the jump from θ to θ'. Denote the multi-set of paths, characterised by these labels, by $\mathcal{L}_{\theta\theta'}$. Examples of elements of $\mathcal{L}_{\theta\theta'}$ are T_1 (if T_1 is pre-enabled in θ and its firing leads to θ'), $T_1 + T_2$ (if there is a non-zero probability that T_1 and T_2 will fire at exactly the same time, and their combined firing leads to θ'), $T_4 \circ T_3$ (if T_3 is pre-enabled in θ, its firing leads to the immediate transition T_4 being enabled, and the firing of T_4 leads to θ'), etc.

Next, we factorise Q by conditioning on the path $L \in \mathcal{L}_{\theta\theta'}$ along which the jump is made. Under the condition that a jump occurs:

$$Q(\theta', A; \theta, x) = \sum_{L \in \mathcal{L}_{\theta\theta'}} p_{\theta', x' | \theta, x, L}(\theta', A \mid \theta, x, L) \times p_{L|\theta, x}(L \mid \theta, x),$$

where $p_{\theta', x' | \theta, x, L}(\theta', A \mid \theta, x, L)$ denotes the conditional probability that the SDCPN state immediately after the jump is in (θ', A), given that the SDCPN state just prior to the jump equals (θ, x), given that the set of transitions L fires to establish the jump. Moreover, $p_{L|\theta,x}(L \mid \theta, x)$ denotes the conditional probability that the set of transitions L fires, given that the SDCPN state immediately prior to the jump equals (θ, x).

In the remainder of this appendix, first $p_{L|\theta,x}(L \mid \theta, x)$ is characterised for each $L \in \mathcal{L}_{\theta\theta'}$. Next, $p_{\theta',x'|\theta,x,L}(\theta', A \mid \theta, x, L)$ is characterised for each $L \in \mathcal{L}_{\theta\theta'}$.

Characterisation of $p_{L|\theta,x}(L \mid \theta, x)$ for each $L \in \mathcal{L}_{\theta\theta'}$

First, assume that $\mathcal{L}_{\theta\theta'}$ does not contain immediate transitions. This yields: each $L \in \mathcal{L}_{\theta\theta'}$ either contains one or more guard transitions, or one delay transition (other combinations occur with zero probability). In particular, $\mathcal{L}_{\theta\theta'}$ is a subset of $\mathcal{B}_\theta$ defined earlier. Then $p_{L|\theta,x}(L \mid \theta, x)$ is determined by $p_{L|\theta,x}(L \mid \theta, x) = \frac{p_L(\theta,x)}{\sum_{B \in \mathcal{L}_{\theta\theta'}} p_B(\theta,x)}$, with $p_B(\theta, x)$ defined earlier.

Next, consider the situations where $\mathrm{RG}_{\theta\theta'}$ may also contain nodes that enable immediate transitions. If L is of the form $L = T_j \circ T_k$, with T_j an immediate transition, then $p_{L|\theta,x}(L \mid \theta, x) = p_{T_k|\theta,x}(T_k \mid \theta, x)$, with the right-hand-side constructed as above for the case without immediate transitions. The same value $p_{T_k|\theta,x}(T_k \mid \theta, x)$ follows for cases like $L = T_m \circ T_j \circ T_k$, with T_j and T_m immediate transitions. However, if the firing of T_k enables more than one immediate transition, then the value of $p_{T_k|\theta,x}(T_k \mid \theta, x)$ is equally divided among the corresponding paths. This means, for example, that if there are $L_1 = T_j \circ T_k$ and $L_2 = T_m \circ T_k$ then $p_{L_1|\theta,x}(L_1 \mid \theta, x) = p_{L_2|\theta,x}(L_2 \mid \theta, x) = \frac{1}{2} p_{T_k|\theta,x}(T_k \mid \theta, x)$.

With this, $p_{L|\theta,x}(L \mid \theta, x)$ is uniquely characterised.

Characterisation of $p_{\theta',x'|\theta,x,L}(\theta', A \mid \theta, x, L)$ for each $L \in \mathcal{L}_{\theta\theta'}$

For probability $p_{\theta',x'|\theta,x,L}(\theta', A \mid \theta, x, L)$, first notice that both (θ, x) and (θ', x') represent states of the complete SDCPN, while the firing of L changes the SDCPN only locally. This yields that in general, several tokens stay where they are when the SDCPN jumps from θ to θ' while the set L of transitions fires.

- $p_{\theta',x'|\theta,x,L}(\theta', A \mid \theta, x, L) = 0$ if for all $x' \in A$, the components of x and x' that correspond with tokens not moving to another place when transitions L fire, are unequal.

In all other cases:

- Assume L consists of one transition T that, given θ and x, is enabled and will fire. Define again c_T^x as the vector containing the colours of the input tokens of T; c_T^x may not be unique. For each c_T^x that can be identified, a sample from $\mathcal{F}_T(\cdot, \cdot; c_T^x)$ provides a vector e' that holds a one for each output arc along which a token is produced and a zero for each output arc along which no token is produced, and it provides a vector c' containing the colours of the tokens produced. These elements together define the size of the jump of the SDCPN state. This gives:

$$p_{\theta',x'|\theta,x,L}(\theta', A \mid \theta, x, L) = \sum_{c^x_T} \int_{(e',c')} \mathcal{F}_T(e', c'; c^x_T) \times \mathbf{I}_{(\theta',A;e',c',c^x_T)},$$

where $\mathbf{I}_{(\theta',A;e',c',c^x_T)}$ is the indicator function for the event that if tokens corresponding with c^x_T are removed by T and tokens corresponding with (e', c') are produced, then the resulting SDCPN state is in (θ', A).

- If L consists of several transitions $T_1, \ldots, T_m$ that, given θ and x, will all fire at the same time, then the firing measure $\mathcal{F}_T$ in the equation above is replaced by a product of firing measures for transitions $T_1, \ldots, T_m$:

$$p_{\theta',x'|\theta,x,L}(\theta', A \mid \theta, x, L) = \sum_{c^x_{T_1},\ldots,c^x_{T_k}} \int_{(e'_1,c'_1),\ldots,(e'_k,c'_k)} \mathcal{F}_{T_1}(e'_1, c'_1; c^x_{T_1}) \times \cdots \times$$

$$\times \mathcal{F}_{T_k}(e'_k, c'_k; c^x_{T_k}) \times \mathbf{I}_{(\theta',A;e'_1,c'_1,c^x_{T_1},\ldots,e'_k,c'_k,c^x_{T_k})},$$

where $\mathbf{I}_{(\theta',A;e'_1,c'_1,c^x_{T_1},\ldots,e'_k,c'_k,c^x_{T_k})}$ denotes indicator function for the event that the combined removal of $c^x_{T_1}$ through $c^x_{T_k}$ by transitions T_1 through T_k, respectively, and the combined production of (e'_1, c'_1) through (e'_k, c'_k) by transitions T_1 through T_k, respectively, leads to a SDCPN state in (θ', A).

- If L is of the form $L = T_j \circ T_k$, with T_j an immediate transition, then the result is:

$$p_{\theta',x'|\theta,x,L}(\theta', A \mid \theta, x, L) = \sum_{c^x_{T_k}} \int_{(e'_j,c'_j,c_j,e'_k,c'_k)} \mathcal{F}_{T_j}(e'_j, c'_j; c_j) \times \mathcal{F}_{T_k}(e'_k, c'_k; c^x_{T_k}) \times$$

$$\times \mathbf{I}_{(\theta',A;e'_j,c'_j,e'_k,c'_k,c^x_T)},$$

where $\mathbf{I}_{(\theta',A;e'_j,c'_j,e'_k,c'_k,c^x_T)}$ denotes indicator function for the event that the removal of $c^x_{T_k}$ and the production of (e'_k, c'_k) by transition T_k leads to T_j having a vector of colours of input tokens c_j and the subsequent removal of c_j and the production of (e'_j, c'_j) by transition T_j leads to a SDCPN state in (θ', A).

- In cases like $L = T_m \circ T_j \circ T_k$, with T_j and T_m immediate transitions, the firing functions of this sequence of transitions are multiplied in a similar way as above.

With this, probability measure Q of the constructed GSHP is uniquely characterised in terms of SDCPN elements.