



NLR-TP-2000-428

**Piecewise Deterministic Markov Processes  
represented by Dynamically Coloured Petri  
Nets**

Revised edition

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



NLR-TP-2000-428

**Piecewise Deterministic Markov Processes  
represented by Dynamically Coloured Petri  
Nets**

Revised edition

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

This research has been carried out under NLR's basic research programme, Work Plan number L.1.B.2.

The contents of this report may be cited on condition that full credit is given to NLR and the authors.

Division:	Air Transport
Issued:	December 1999
Classification of title:	Unclassified



### DOCUMENT CONTROL SHEET

Report Title	Piecewise Deterministic Markov Processes represented by Dynamically Coloured Petri Nets		
Version	0.3		
Date	21-03-2003		
Customer	NLR	Order No.	5091922
Project	DYNAMO	NLR division	Air Transport

### Document change log

Version Nr	Issue date	Details of changes
0.1	28-12-1999	IEEE anonymous peer review comments incorporated
0.2	15-09-2000	Review comments of NLR reviewer incorporated
0.3	21-03-2003	Technical update

Version 0.3		Signature	Date
Authors	M.H.C. Everdij		
	H.A.P. Blom		
Internal reviewer	M.B. Klompstra		



## Summary

Piecewise Deterministic Markov Processes (PDPs) are known as the largest class of Markov processes virtually describing all continuous-time processes not involving diffusions. In general the state space of a PDP is of hybrid type, i.e. a tensor 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 Petri Nets that can play a similar role for developing PDP models for complex practical problems. To fulfil this need, the paper defines a new type of Petri Net, named Dynamically Coloured Petri Net (DCPN), and proves that there exist into-mappings between PDPs and DCPNs.

This report version (dated 2003) is a slightly revised edition of a report (dated 2000) with the same title, authors and report number.



## Contents

<b>1</b>	<b>Introduction</b>	8
<b>2</b>	<b>Piecewise Deterministic Markov Processes</b>	9
<b>3</b>	<b>Main Petri Nets extensions</b>	11
<b>4</b>	<b>Dynamically Coloured Petri Nets</b>	12
<b>5</b>	<b>Example DCPN</b>	15
<b>6</b>	<b>Piecewise Deterministic Markov Processes into Dynamically Coloured Petri Nets</b>	17
<b>7</b>	<b>Dynamically Coloured Petri Nets into Piecewise Deterministic Markov Processes</b>	20
<b>8</b>	<b>Example PDP</b>	23
<b>9</b>	<b>Conclusions</b>	24
<b>10</b>	<b>References</b>	24
	 1 Figure	
	 <b>Appendices</b>	27
<b>A</b>	<b>Formal definition of Dynamically Coloured Petri Nets</b>	27
<b>B</b>	<b>Characterisation of <math>Q</math> in terms of DCPN</b>	30

(32 pages in total)

### List of acronyms

DCPN	Dynamically Coloured Petri Net
ECPN	Extended Coloured Petri Net
HLHPN	High-Level Hybrid Petri Net
PDP	Piecewise Deterministic Markov Process
PN	Petri Net
RG	Reachability Graph
RRG	Reduced Reachability Graph

### List of symbols

$\cup_{\theta}$	: Unity
$\cap_{\theta}$	: Intersection
$ \cdot $	: Number of elements in a set
$\mathbb{R}^n$	: $n$ -dimensional rational numbers
$\mathbb{N}$	: Natural numbers
$\partial E$	: Boundary of open subset $E$
$R_0 - R_4$	: Rules
$D_1 - D_3$	: Conditions
$x_t$	: Continuous state
$\theta_t$	: Discrete state
$\xi_t$	: Hybrid state
$E_{\theta}$	: Open subset of $\mathbb{R}^{d(\theta)}$
$E$	: Disjoint unity of all subsets $E_{\theta}$
$\mathcal{E}$	: Borel-measurable subsets of $E$
$\Gamma^*$	: Reachable boundary of $E$
$\mathbf{K}$	: Countable domain for process $\{\theta_t\}$
$d(\cdot)$	: Function that maps $\mathbf{K}$ into $\mathbb{N}$
$g_{\theta}(\cdot)$	: Lipschitz continuous function
$\phi_{\theta, x_0}$	: Flow
$\lambda(\theta_t, x_t)$	: Rate of Poisson point process
$G_{\xi}(\cdot)$	: Survivor function
$I_A$	: Indicator function
$Q(\cdot; \xi)$	: Transition measure

$t$	: Time
$\tau, \tau_k$	: Stopping times
$\Delta$	: Time
$t_*(\theta, x)$	: Time until first boundary hit
$t_\infty(\theta, x)$	: Explosion time of flow $\phi_{\theta, x}(\cdot)$
$\sigma_k, \zeta_k$	: Samples from probability distributions
$N_t$	: Number of jumps until time $t$
$\mathcal{P}$	: Set of places
$\mathcal{P}(A)$	: The place that is connected to arc $A$
$\mathcal{T}$	: Set of transitions
$\mathcal{T}_G$	: Set of guard transitions
$\mathcal{T}_D$	: Set of delay transitions
$\mathcal{T}_I$	: Set of immediate transitions
$\mathcal{T}(A)$	: The transition that is connected to arc $A$
$\mathcal{A}$	: Set of arcs
$\mathcal{A}_O$	: Set of ordinary arcs
$\mathcal{A}_E$	: Set of enabling arcs
$\mathcal{A}_I$	: Set of inhibitor arcs
$A(T)$	: Set of arcs connected to transition $T$
$A_{in}(T)$	: Set of input arcs of transition $T$
$A_{in,O}(T)$	: Set of ordinary input arcs of transition $T$
$A_{in,OE}(T)$	: Set of input arcs of transition $T$ that are either ordinary or enabling
$A_{out}(T)$	: Set of output arcs of transition $T$
$\mathcal{N}$	: Node function
$\mathcal{S}$	: Set of colour types
$\mathcal{C}$	: Colour function
$\mathcal{I}$	: Initial marking
$\mathcal{V}$	: Set of token colour functions
$\mathcal{V}_P$	: Token colour function for place $P$
$\mathcal{G}$	: Set of transition guards
$\mathcal{G}_T$	: Transition guard for transition $T$
$\mathcal{D}$	: Set of transition delays
$\mathcal{D}_T$	: Transition delay for transition $T$




---

$\mathcal{F}$	:	Set of firing measures
$\mathcal{F}_T$	:	Firing measure for transition $T$
$P, P_i$	:	Place
$P(A(T))$	:	Set of places connected to $T$ by the set of arcs $A(T)$
$T, T_i$	:	Transition
$T_i^G$	:	Guard transition
$T_i^D$	:	Delay transition
$A, A_i$	:	Arc
$\mathcal{C}(P)_{ms}$	:	Set of all multisets over $\mathcal{C}(P)$
$c, c_t$	:	Colour of token or vector of colours
$\delta_T$	:	Rate of transition delay
$z_t$	:	Vector containing position and velocity of aircraft
$v_t$	:	Velocity of aircraft
$u$	:	Random number
$p_{B C}$	:	Conditional probability density function
$\alpha_\ell$	:	Auxiliary variable
$f$	:	Vector of zeros and ones
$\vartheta_i, m_i$	:	Element of $\mathbf{K}$
$v_{i,t}, v_i$	:	Number of tokens in place $P_i$
$V, V^i$	:	Nodes of reachability graph
$x_{i,j,t}$	:	Colour of $j$ th token in place $P_i$ at time $t$
$\mathcal{L}$	:	Set of paths characterised by labels
$L$	:	Set of transitions, element of $\mathcal{L}$





## 1 Introduction

The motivation for this research stems from the following unsolved issue in air traffic: under which conditions is it possible to reduce established criteria for separation between aircraft without sacrificing safety in the form of collision risk. Studying this issue is most urgent for those regions in the world where air traffic is most dense, and consequently where the interplay between aircraft and air traffic management centres is most complex.

During an earlier study (Bakker and Blom, 1993) a clear relation has been established between collision risk and the evolution of the density of the joint states of two or more flying aircraft. During the subsequent search for Markov process models to characterise the evolution of such densities the class of Piecewise Deterministic Markov Processes (PDPs) was identified as a very useful one (Everdij *et al.*, 1996). At this moment, this type of modelling and evaluation has been accomplished for several air traffic management situations (Blom *et al.*, 1998). It appeared to be less straightforward to develop an appropriate PDP model for a process as complex as air traffic management. For this reason a Petri Net has been developed (Everdij *et al.*, 1997) that supports the modelling of PDPs for complex practical problems, similarly as Stochastic Petri Nets support the development of a Markov Chain for discrete valued complex problems. This new Petri Net and its precise relation with PDPs form the subject of this paper.

Davis (1984, 1993) has introduced PDPs as the most general class of continuous-time Markov processes which include both discrete and continuous processes, except diffusion. A PDP consists of two components: a discrete valued component and a continuous valued component. The discrete valued component models the mode process  $\{\theta_t\}$ . At discrete times,  $\{\theta_t\}$  may switch to another mode value which is selected according to some probabilistic relation. The continuous valued component models the drift process  $\{x_t\}$ , as a solution of a  $\theta_t$ -dependent differential equation. At discrete moments in time,  $\{x_t\}$  may jump according to some relation, which makes it only piecewise continuous. The PDP state is given by  $\xi_t = \text{Col}\{\theta_t, x_t\}$ , and is called a *hybrid state*. A switch and/or a jump occurs either when a doubly stochastic Poisson process generates a point or when  $\{x_t\}$  hits the boundary of a predefined area. If  $\{x_t\}$  also makes a jump at a time when  $\{\theta_t\}$  switches, this is said to be a hybrid jump. PDPs are defined such that their sample paths are right-continuous and have left-hand-side limits, often shortened as càdlàg, which is an acronym for the French ‘continu à droite, limites à gauche’, see e.g. Protter (1990).

Branicky (1995) identified a close relation between PDPs and Hybrid Automata. The latter have been shown useful for application in problems of decidability, formal verification and control synthesis (Alur *et al.*, 1993; Lygeros *et al.*, 1998; van Schuppen, 1998; Sipser, 1997; Tomlin *et al.*, 1998; Weinberg *et al.*, 1996). An important limitation, however, is that Poisson type of events

of PDPs are not covered by Hybrid Automata, which makes them less effective in modelling stochastic effects than PDPs.

A Petri Net representing a PDP should make direct use of the specific PDP structure in order to have an advantageous modelling power. The established Petri Net descriptions (see David and Alla (1994) for a good overview) appear not to have this property. The newly developed Dynamically Coloured Petri Net (DCPN) presented in this paper does. The values of the mode process can be mapped to the distribution of tokens among the places in the DCPN. As long as the tokens remain where they are, the mode keeps the same value. The continuous process values are directly coupled to the token colours but they are evaluated while the tokens reside in the places and not during the transport of tokens. These colours may however influence the mode switches, both in time and in value. In this way, the continuous valued processes and the discrete valued processes are coupled and decoupled at the same level as they are in Piecewise Deterministic Markov Processes. This makes that into-mappings between PDPs and DCPNs exist. The key result of this paper is the proof of the existence of these into-mappings. An issue that deserves special attention when relating PDPs to Petri Nets is that for PDP, at each moment in time, there is a unique realisation of the state, while a Petri Net may make a sequence of jumps at a single moment in time. The into-mappings between PDPs and DCPNs we develop in this paper take care of this issue.

The organisation of this paper is as follows. First, in Section 2, Piecewise Deterministic Markov Processes are described briefly. Next, Section 3 discusses how the main Petri Nets extensions from literature relate to PDPs. Subsequently, Section 4 defines Dynamically Coloured Petri Nets. Section 5 presents a DCPN example for a practical situation. Section 6 shows that a PDP can be represented by a DCPN. Section 7 shows that a DCPN can be represented by a PDP. Section 8 gives an example PDP that models the same process as the one described by the example DCPN in Section 5. Finally, Section 9 draws conclusions.

## 2 Piecewise Deterministic Markov Processes

A Piecewise Deterministic Markov Process is defined as follows (see Davis (1993)): For each  $\theta$  in its countable domain  $\mathbf{K}$ , let  $E_\theta$  be an open subset<sup>1</sup> of  $\mathbb{R}^{d(\theta)}$ , where  $d$  is a function that maps  $\mathbf{K}$  into  $\mathbb{N}$ . For each  $\theta \in \mathbf{K}$ , consider the ordinary differential equation  $\dot{x}_t = g_\theta(x_t)$ , with  $x_0 \in E_\theta$ , where  $g_\theta : \mathbb{R}^{d(\theta)} \rightarrow \mathbb{R}^{d(\theta)}$  is a locally Lipschitz continuous function. Given the initial value  $x_0$ , this differential equation has a unique solution given by the flow  $\phi_{\theta, x_0}$ . Suppose at some time instant  $\tau$  the PDP state  $\xi_\tau$  assumes value  $\xi = (\theta, x) = (\theta, \phi_{\theta, x}(0))$ . If no jumps occur, the

---

<sup>1</sup>Note that Davis writes  $E_\theta = E_\theta^0 \cup \partial_1 E_\theta^0$ , with  $E_\theta^0$  an open subset of  $\mathbb{R}^{d(\theta)}$ , and  $\partial_1 E_\theta^0$  those points on the boundary of  $E_\theta^0$  from which  $E_\theta^0$  can be reached (by the flow  $\phi$ ), but which cannot be reached from the interior of  $E_\theta^0$ .



process state at  $t \geq \tau$  is given by  $\xi_t = (\theta_t, x_t) = (\theta, \phi_{\theta, x}(t - \tau))$ . At some moment in time, however, the process 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  $\partial E_\theta$  of  $E_\theta$ ,  $t > \tau$ .

This yields for the survivor function of the first jump (i.e. the probability that the first jump occurs at least  $t - \tau$  time units after  $\tau$ ), starting from point  $\xi = (\theta, x)$  at  $t = \tau$ :

$$G_\xi(t - \tau) \triangleq I_{(t - \tau < t_*(\theta, x))} \cdot e^{-\int_\tau^t \lambda(\theta, \phi_{\theta, x}(s - \tau)) ds}, \quad (1)$$

where  $I$  is an indicator function and  $t_*(\theta, x)$  denotes the time until the first boundary hit after  $t = \tau$ , which is given by  $t_*(\theta, x) \triangleq \inf\{t - \tau > 0 \mid \phi_{\theta, x}(t - \tau) \in \partial E_\theta\}$ . The first factor in (1) is explained by the boundary hitting process: after the process state has hit the boundary, which is when  $t - \tau = t_*(\theta, x)$ , this first factor ensures that the survivor function evaluates to zero. The second factor in (1) comes from the Poisson process: this second factor ensures that a jump is generated after an exponentially distributed time with a rate  $\lambda$  that is dependent on the PDP state. The time until the first jump is calculated by drawing a sample from  $G_\xi(\cdot)$ . Also available is a 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.

With this, the algorithm to determine a sample path for the hybrid state process  $\xi_t$ ,  $t \geq 0$ , from the initial state  $\xi_0 = (\theta_0, x_0)$  on, is in two 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  $k = 1, 2, \dots$ :

**Step 1:** Draw a sample  $\sigma_k$  from survivor function  $G_{\xi_{\tau_{k-1}}}(\cdot)$ , then the time  $\tau_k$  of the  $k$ th jump is  $\tau_k = \tau_{k-1} + \sigma_k$ . The sample path up to the  $k$ th 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  $\zeta_k$  from transition measure  $Q(\cdot; \xi'_{\tau_k})$ , where  $\xi'_{\tau_k} = (\theta_{\tau_{k-1}}, \phi_{\theta_{\tau_{k-1}}, x_{\tau_{k-1}}}(\tau_k - \tau_{k-1}))$ . Then, if  $\tau_k < \infty$ , the process state at the time  $\tau_k$  of the  $k$ th jump is given by

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

Following Section 24.8 of Davis (1993), the PDP conditions are:

- $C_1$   $g_\theta$  is a locally Lipschitz continuous function, which, for each initial state  $(\theta, x)$ , determines a flow  $\phi_{\theta, x}(\cdot)$ . If  $t_\infty(\theta, x)$  denotes the explosion time of the flow  $\phi_{\theta, x}(\cdot)$ , i.e.  $|\phi_{\theta, x}(t)| \rightarrow \infty$  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  $E = \cup_{\theta} E_{\theta}$ ,  $\lambda : E \rightarrow \mathbb{R}^+$  is a measurable function such that for all  $\xi \in E$ , there is  $\epsilon(\xi) > 0$  such that  $t \rightarrow \lambda(\theta, \phi_{\theta,x}(t))$  is integrable on  $[0, \epsilon(\xi)[$ .
- $C_3$  With  $E$  as above and  $\Gamma^*$  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  $\xi \rightarrow Q(A; \xi)$  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  $\xi$  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.

### 3 Main Petri Nets extensions

This section discusses how the main Petri Nets extensions from literature relate to PDPs.

One interesting Petri Net extension is named Hybrid Petri Net (Le Bail *et al.*, 1991), which is a generalisation of Continuous Petri Net (David and Alla, 1987). Besides places that may contain discrete tokens, the Hybrid Petri Net has places that may contain a non-negative real-valued amount of tokens. Transitions connected to these places consume continuous amounts of these tokens at certain quantities per time unit and next produce these amounts for other continuous places. The state of the Hybrid Petri Net at each time instant  $t$  is written as a vector giving for each place its marking, i.e. the amount of tokens it contains. The marking of a place at a later time instant is equal to the marking at time  $t$ , minus the amount of token consumed by output transitions of the place, plus the amount produced by input transitions of the place. Since a change of marking for one place automatically incorporates a change of marking for another place, and since all amounts of tokens should be non-negative, into-mappings between PDPs and Hybrid Petri Nets are far from obvious.

Related to Hybrid Petri Net is Fluid Stochastic Petri Net (Trivedi and Kulkarni, 1993), which also moves fluid tokens between continuous places and discrete tokens between discrete places. The transition times for the discrete tokens are exponentially distributed (possibly depending on the fluid level) such that the discrete Petri Net part gives rise to a continuous time Markov process. The discrete token marking influences the continuous flow rate between the continuous places. In the continuous net conservation of token mass is to be preserved. Hence, into-mappings between PDPs and Fluid Stochastic Petri Nets are generally not feasible.

In Extended Coloured Petri Net (ECPN), as introduced by Yang *et al.* (1995), the token colours are real-valued vectors that may follow the solution path of a difference equation. The token colour is updated in an external loop around its residence place by an additional updating transition. The state of the ECPN is given by the current colours of all tokens and the places they reside in. PDPs might be represented by ECPN but only in an approximate way: the mode values

might be mapped to the places and the drift process might be mapped to the colours of the tokens; however, the continuous-time drift process component has to be approximated by means of a discrete-time difference equation. This makes that a boundary hitting can never be exactly timed. Moreover, the necessary presence of updating transitions may result in a very large Petri Net graph when modelling a complex process.

High-Level Hybrid Petri Net (HLHPN) has been introduced by Giua and Usai (1996) as a further elaboration of Hybrid Petri Net. In an HLHPN there are two kinds of places: the usual places for discrete tokens and a new type of places storing real-valued tokens which follow the solution path of a differential equation. A token switch between discrete places may generate a jump in the value of the real valued vector. Advantage of this extension with respect to Hybrid Petri Net is that continuous valued processes can be modelled that do not need to have a positive value. Advantage with respect to ECPN is the avoidance of the time discretisation. Similarly as with ECPN, HLHPNs use updating transitions which still may result in a very large Petri Net graph when modelling a complex process.

Related to HLHPN are Differential Petri Nets (Demongodin and Koussoulas, 1998), which have discrete places (having a non-negative integer marking), differential places (having a real valued marking, which can also be negative), discrete transitions and differential transitions. A differential transition is enabled if its discrete input places contain a number of tokens that is larger than or equal to the weight of the corresponding arcs. An enabled differential transition that fires yields a change of marking equal to the speed of the transition, times the weight of the corresponding arc. This speed may be a constant, a linear combination, or a non-linear function of the markings connected to the transition (the speed may also be negative). The facts that time is discretised and that the markings are one-dimensional, may result in a very large Petri Net graph when modelling a complex process. Moreover, mode switches are enabled by integer bounds only.

This overview leads to the conclusion that there are many valuable Petri Net extensions available in literature that are related to PDPs; however, for none of these the existence of into-mappings with PDPs has been proven.

#### **4 Dynamically Coloured Petri Nets**


This section presents a definition of Dynamically Coloured Petri Net. Since the formal definition is notationally cumbersome, in this section only an informal description is given. In Appendix A the formal definition is described.



**Definition 1:**



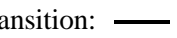
A Dynamically Coloured Petri Net (DCPN) is an 11-tuple  $DCPN = (\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{C}, \mathcal{I}, \mathcal{V}, \mathcal{G}, \mathcal{D}, \mathcal{F})$ , together with some rules  $R_0 - R_4$ , where:

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



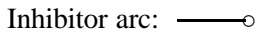
Place: 

2.  $\mathcal{T}$  is a finite set of transitions, which consists of three disjoint sets: 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.

Notations are:

Guard transition:  Delay transition:  Immediate transition: 

3.  $\mathcal{A}$  is a finite set of arcs, which consists of three disjoint sets: 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. Notations are:

Ordinary arc:  Enabling arc:  Inhibitor arc: 

4.  $\mathcal{N}$  is a node function, which uniquely maps each arc  $A$  in  $\mathcal{A}$  to an ordered pair of one transition and one place. Enabling arcs and inhibitor arcs always point from a place to a transition, ordinary arcs may point from a place to a transition or from a transition to a place. If a pair of nodes  $(P, T)$  is directly connected by an inhibitor arc, it cannot also be directly connected by another arc.

Components  $\mathcal{P}, \mathcal{T}, \mathcal{A}$  and  $\mathcal{N}$  determine the DCPN graph. A place is *current* if a *token* (denoted by a dot  $\bullet$ ) resides in it. Multiple tokens may reside in the DCPN, even inside one place. In a DCPN, the tokens may have values, also named *colours*. These colours can be of particular types only. The set of allowed token colour types is denoted by  $\mathcal{S}$ . The function that maps each place to such type is denoted by  $\mathcal{C}$ .

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.
6.  $\mathcal{C}$  is a colour function, which maps each place  $P \in \mathcal{P}$  to a specific colour type in  $\mathcal{S}$ . Each token residing in place  $P$  must have a colour of type  $\mathcal{C}(P)$ .

The set of tokens initially residing in the DCPN, together with their colours, is specified by component  $\mathcal{I}$ .

7.  $\mathcal{I}$  defines the initial marking of the net, i.e., for each place  $P \in \mathcal{P}$  it specifies the number of tokens (possibly zero) initially in it, together with the colours these tokens have (these colours must be of type  $\mathcal{C}(P)$ ). The initial marking is such, that no immediate transition is immediately enabled.

In DCPN, the colour of a token may change through time while the token is residing in its place. This behaviour is described by component  $\mathcal{V}$  below.

8.  $\mathcal{V}$  is a set of token colour functions. For each place  $P \in \mathcal{P}$  it contains a locally Lipschitz continuous function  $\mathcal{V}_P$ . If a token in place  $P$  has colour  $c$  at time  $\tau$ , and if it stays in that place up to time  $t > \tau$ , then the colour  $c_t$  at time  $t$  equals the unique solution of the differential equation  $\dot{c}_t = \mathcal{V}_P(c_t)$  with initial condition  $c_\tau = c$ .

Tokens can be removed from places by transitions that are connected to these places by incoming ordinary arcs. A transition can only remove tokens if two conditions are both satisfied. If this is the case, the transition is said to be *enabled*. The first condition is that the transition must have at least one token per ordinary arc and one token per enabling arc in each of its input places and have no token in the input places to which it is connected by an inhibitor arc. When the first condition holds, the transition is said to be *pre-enabled*. The second condition differs per type of transition. For immediate transitions the second condition is automatically satisfied if the transition is pre-enabled. For guard transitions the second condition is specified by the set of transition guards  $\mathcal{G}$  and for delay transitions it is specified by the set of transition delays  $\mathcal{D}$ . The evaluation of  $\mathcal{G}$  and  $\mathcal{D}$  may depend on a vector of colours of input tokens. This vector is formed by taking one token per incoming ordinary arc and enabling arc from the input places of the transition and listing their colours in one vector. If this vector is not unique (for example, if one input place contains several tokens per arc), all possible vectors are evaluated in parallel and the transition is enabled if one of these vectors satisfies the second condition. The value of the vector(s) may still change through time according to the corresponding token colour functions. Both  $\mathcal{G}$  and  $\mathcal{D}$  are described below:

9.  $\mathcal{G}$  is a set of transition guards. For each  $T \in \mathcal{T}_G$ , it contains a boolean-valued transition guard  $\mathcal{G}_T$ . The guard of  $T$  evaluates to True when the vector containing the colours of the input tokens (see above) enters a  $T$ -specific boundary  $\partial G_T$  for the first time. When the guard evaluates to True, the second condition holds and the transition is enabled. Guard evaluation of transition  $T$  immediately stops when  $T$  is not pre-enabled anymore; evaluation is restarted when it is.
10.  $\mathcal{D}$  is a set of transition delays. For each  $T \in \mathcal{T}_D$ , it contains a transition delay  $\mathcal{D}_T$ , which specifies the time between when  $T$  is pre-enabled and when it is enabled. This waiting time is exponentially distributed, with a rate  $\delta_T$  which is an integrable function of the vector of colours of the input tokens (see above). As for the guard, the delay evaluation immediately stops when  $T$  is not pre-enabled anymore; evaluation is restarted when it is.

When a transition is enabled, it removes one token along each ordinary arc from its input places (rules  $R_1$  and  $R_2$  below are used if this token is not unique, e.g., if several vectors of input colours have been evaluated in parallel and two of them enable  $T$  at exactly the same time).

Tokens are not removed along enabling arcs and should not be existent in a place connected to  $T$



by an inhibitor arc.

Next, at time  $\tau$ ,  $T$  produces zero or one token along each output arc (note that output arcs are always ordinary arcs). The colours of the tokens produced and the arcs along which they are produced are specified by component  $\mathcal{F}$ :

11.  $\mathcal{F}$  is a set of probability measures, named firing measures. For each  $T \in \mathcal{T}$  it contains a probability measure  $\mathcal{F}_T$  which is assumed continuous. If  $c_\tau$  denotes the vector of colours of tokens that enabled  $T$ , then a sample from  $\mathcal{F}_T(\cdot, \cdot; c_\tau)$  contains a vector  $f$  of zeros and ones, and a vector  $c$  of output token colours. The vector  $f$  holds a 1 for each output arc of  $T$  along which a token is produced, and a zero for each output arc of  $T$  along which no token is produced. Vector  $c$  contains the colours of the corresponding new tokens. The colours of the new tokens have sample paths that start at time  $\tau$ .

Finally, there are some rules  $R_0 - R_4$ :

12.  $R_0$  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 (i.e. each set has an equal probability to be selected).
- $R_3$  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 considered 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.

**Remark:**

The firing of any transition "stops the time", i.e. if a transition becomes enabled at time  $\tau$ , it also fires tokens in its output places at time  $\tau$ . And if, for example, there is a sequence of immediate transitions that enable each other with their firing, then their consecutive firings all take place at this same time instant  $\tau$ .

With this, the (informal) DCPN definition ends. For a more formal definition, see Appendix A.

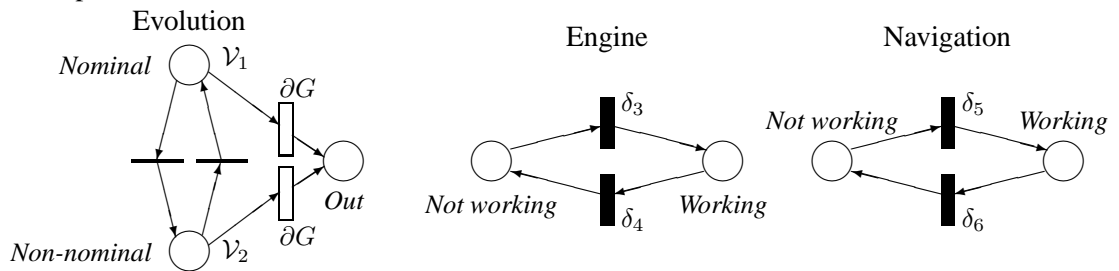


## 5 Example DCPN

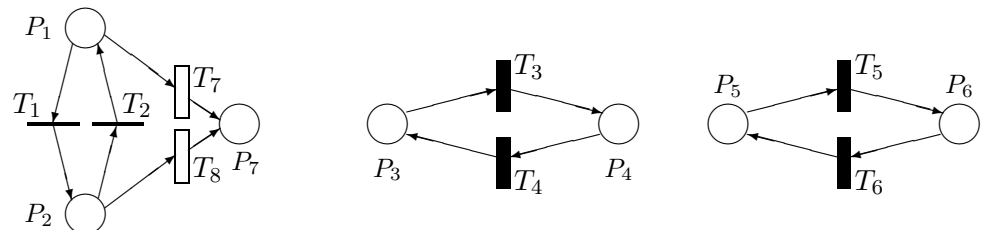
To illustrate the advantages of DCPN when modelling a complex system, consider a very simplified model of the evolution of an aircraft in one sector of airspace.

Assume the deviation of this aircraft from its intended path depends on the operability 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  $\delta_3$  (engine repaired),  $\delta_4$  (engine fails),  $\delta_5$  (navigation repaired) and  $\delta_6$  (navigation fails), respectively. The operability of these systems has the following effect on the aircraft path: if both systems are *Working*, the rate of change of the position and velocity of the aircraft is given by function  $\mathcal{V}_1$  (i.e. if  $z_t$  is a vector containing this position and velocity then  $\dot{z}_t = \mathcal{V}_1(z_t)$ ). If either one, or both, of the systems is *Not working*, the rate of change of the position and velocity of the aircraft is given by  $\mathcal{V}_2$ . 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 crosses the boundary  $\partial G$  to a neighbouring airspace sector.

A DCPN instantiation for this example can be constructed in three phases. In the first phase the local Petri Net graphs (local PNs) are instantiated, based on the modes identified above. The result is presented below:

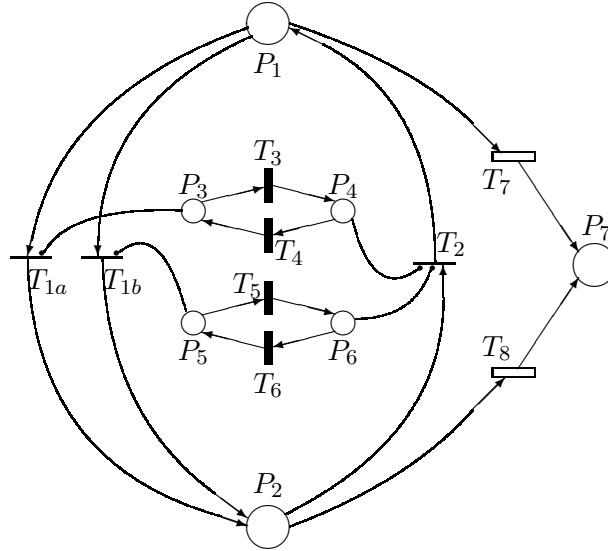


Or, with all places and transitions given abbreviated names:



In the second phase of constructing the DCPN instantiation, the effect of the Engine and Navigation local PN on the Evolution local PN is made explicit by coupling the local PNs by additional arcs (and, if necessary, additional places or transitions). Here, removal of a token from

one local PN by a transition of another local PN is prevented by using enabling arcs instead of ordinary arcs. The resulting graph is presented below. Notice that transition  $T_1$  is replaced by two transitions  $T_{1a}$  and  $T_{1b}$ .



The graph above completely defines DCPN 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  $\mathcal{T}_I = \{T_{1a}, T_{1b}, T_2\}$ . In the third phase, the other DCPN elements are specified (note that in practice, phases 2 and 3 are applied iteratively until a satisfactory result is obtained):

$\mathcal{S}$ : One colour type is defined;  $\mathcal{S} = \{\mathbb{R}^6\}$ .

$\mathcal{C}$ :  $\mathcal{C}(P_1) = \mathcal{C}(P_2) = \mathcal{C}(P_7) = \mathbb{R}^6$ . 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  $P_6$ , no colour type needs to be defined (one might define a dummy colour type for these places, but this is not pursued further).

$\mathcal{I}$ : Place  $P_1$  initially has a token with colour  $z_0 = (x_0, v_0)' \in \mathbb{R}^6$ . Places  $P_4$  and  $P_6$  initially each have a token with no colour.

$\mathcal{V}$ : The token colour functions for places  $P_1$ ,  $P_2$  and  $P_7$  are defined by  $\mathcal{V}_{P_1} = \mathcal{V}_1$ ,  $\mathcal{V}_{P_2} = \mathcal{V}_2$  and  $\mathcal{V}_{P_7} = 0$ . For places  $P_3 - P_6$  the token colour function is not applicable.

$\mathcal{G}$ : Transitions  $T_7$  and  $T_8$  have a guard that is defined by  $\partial G_{T_7} = \partial G_{T_8} = \partial G \times \mathbb{R}^3$ .

$\mathcal{D}$ : The jump 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 a token with a colour (if applicable) equal to the colour of the token removed, i.e. for all  $T$ ,  $\mathcal{F}_T(1, \cdot; \cdot) = 1$ .

The resulting DCPN is easily understood and verified, mainly because it is constructed in several (in practice, iterative) phases, which is constructive if the process to be modelled is very complex.

In the first phase, local Petri Net graphs are constructed. In the second phase these local graphs are connected into one graph. In the third phase the non-graphical DCPN elements are specified.

## 6 Piecewise Deterministic Markov Processes into Dynamically Coloured Petri Nets

This section shows that each Piecewise Deterministic Markov Process can be represented by a Dynamically Coloured Petri Net, by providing an into-mapping from PDP into the set of DCPNs. Notice that we do not claim the into-mapping to be unique; there may be other DCPN instantiations describing the same PDP.

### Theorem 1:

Each Piecewise Deterministic Markov Process with a finite domain  $\mathbf{K}$  can be represented by a Dynamically Coloured Petri Net.

### Proof:

The proof is completed by considering an arbitrary PDP, and constructing a DCPN that represents it. Consider an arbitrary PDP described by a finite set  $\mathbf{K}$ ,  $\theta_t$ ,  $x_t$ ,  $d(\theta)$ ,  $\xi_0$ ,  $\partial E_\theta$ ,  $g_\theta$ ,  $\lambda$  and  $Q$ . A DCPN representing this PDP is first described intuitively, next all DCPN elements are specified in detail.

In the DCPN instantiation we constructed to represent the PDP, only one token resides. Figure 1 presents a small part of this DCPN. The space of the mode process is assumed to be  $\mathbf{K} = \{\vartheta_1, \dots, \vartheta_{|\mathbf{K}|}\}$ . The figure shows the situation at some time  $\tau_{k-1}$ , when the PDP is given by  $(\theta_{\tau_{k-1}}, x_{\tau_{k-1}})$ . The token resides in place  $P_{\vartheta_i}$ , 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 determined in two steps:

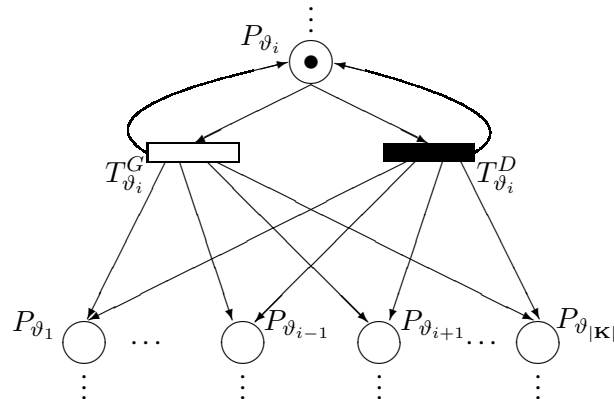


Fig. 1 Part of a Dynamically Coloured Petri Net representing a Piecewise Deterministic Markov Process.

**Step 1:** While the token is residing in place  $P_{\vartheta_i}$ , its colour  $x_t$  changes according to the flow



$\phi_{\vartheta_i, x_{\tau_{k-1}}}$ , i.e.,  $x_t = \phi_{\vartheta_i, x_{\tau_{k-1}}}(t - \tau_{k-1})$ . Transitions  $T_{\vartheta_i}^G$  and  $T_{\vartheta_i}^D$  are both pre-enabled and compete for this token which resides in their common input place  $P_{\vartheta_i}$ . Transition  $T_{\vartheta_i}^G$  models the boundary hitting generating a mode switch, while transition  $T_{\vartheta_i}^D$  models the Poisson process generating a mode switch. The transition that is enabled first, determines the kind of switch occurring. The time at which this happens is denoted by  $\tau_k$ .

**Step 2:** Next, with one of the transitions enabled, its firing measure is evaluated. This firing measure is such, that if a sample  $\zeta_k$  from transition measure  $Q(\cdot; \vartheta_i, \phi_{\vartheta_i, x_{\tau_{k-1}}}(t - \tau_{k-1}))$ , 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  $P_{\vartheta_j}$ . The other places get no token.

After this, the process starts again in the same way from the new state on.

On the basis of this representation, all DCPN elements  $\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{C}, \mathcal{I}, \mathcal{V}, \mathcal{G}, \mathcal{D}, \mathcal{F}$  and the rules  $R_0 - R_4$  are characterised in terms of the PDP as follows:

**P:** The set of places contains  $|\mathbf{K}|$  elements  $P_\theta, \theta \in \mathbf{K}$ .

**T:** The set of transitions contains for each  $\theta \in \mathbf{K}$  two elements  $T_\theta^G$  and  $T_\theta^D$ , with  $T_\theta^G \in \mathcal{T}_G$  and  $T_\theta^D \in \mathcal{T}_D$ . The set  $\mathcal{T}_I$  of immediate transitions is empty.

**A:** There are no inhibitor or enabling arcs:  $\mathcal{A}_I = \emptyset$  and  $\mathcal{A}_E = \emptyset$ , while  $\mathcal{A}_O$  contains  $2|\mathbf{K}| + 2|\mathbf{K}|^2$  ordinary arcs.

**N:** The node function maps each arc in  $\mathcal{A}$  to a pair of nodes. For each element  $\theta \in \mathbf{K}$ , there are two of these pairs:  $(P_\theta, T_\theta^G)$  and  $(P_\theta, T_\theta^D)$ , and additionally for each ordered combination  $(\theta, \vartheta)$ , with  $\theta, \vartheta \in \mathbf{K}$ , there are two pairs of nodes  $(T_\theta^G, P_\vartheta)$  and  $(T_\theta^D, P_\vartheta)$ .

**S:** The set of colour types contains the elements  $\mathbb{R}^{d(\theta)}$ , where  $\theta$  runs through  $\mathbf{K}$ .

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

**I:** All places in the DCPN initially contain zero tokens, except for place  $P_{\theta_0}$ , which contains one token with colour  $x_0$ .

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

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

**D:** For all  $\theta \in \mathbf{K}, \delta_{T_\theta^D}(\cdot) = \lambda(\theta, \cdot)$ .

**F:** If  $x$  denotes the colour of the token removed from place  $P_\theta, (\theta \in \mathbf{K})$ , 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  $\theta \in \mathbf{K}, \mathcal{F}_{T_\theta^D} = \mathcal{F}_{T_\theta^G}$ .

**$R_0 - R_4$ :** Since there are no immediate transitions in the constructed DCPN instantiation, rule  $R_0$  holds true. Since there is only one token in the constructed DCPN instantiation,  $R_1 - R_3$  also hold true. Rule  $R_4$  is in effect when for particular  $\theta$ , transitions  $T_\theta^G$  and  $T_\theta^D$  become enabled at exactly the same time. Since  $\lambda$  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, rule  $R_4$  always holds.

This shows that for any PDP we are able to construct a DCPN instantiation representing it, which completes the proof of Theorem 1.

**Remark 1:**

The DCPN instantiation defined above has many places, and only one token. An interesting problem is to find another into-mapping, in which the DCPN instantiation has fewer places and more tokens. Addressing this problem falls outside the scope of this paper.

## 7 Dynamically Coloured Petri Nets into Piecewise Deterministic Markov Processes

Under some conditions, each Dynamically Coloured Petri Net can be represented by a Piecewise Deterministic Markov Process. In this section this is shown by providing an into-mapping from DCPN into the set of PDPs. Notice that we do not claim the into-mapping to be unique; there may be other PDP instantiations describing the same DCPN.

**Theorem 2:**

Each Dynamically Coloured Petri Net can be represented by a Piecewise Deterministic Markov Process, if conditions  $D_1 - D_3$  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 equals infinity.
- $D_2$  After a transition firing (or after a sequence of firings that occur at that 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.

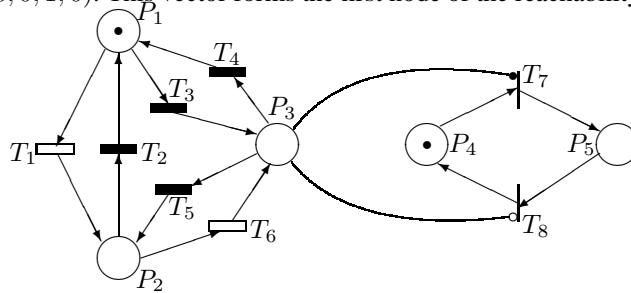
**Proof:**

The proof is completed by considering an arbitrary DCPN that satisfies conditions  $D_1 - D_3$ , and constructing a PDP that represents it. Consider an arbitrary DCPN described by the tuple  $(\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{C}, \mathcal{I}, \mathcal{V}, \mathcal{G}, \mathcal{D}, \mathcal{F})$  and the rules  $R_0 - R_4$ , and assume that the conditions  $D_1 - D_3$  are satisfied. In order to represent this DCPN by a PDP, all PDP elements  $\mathbf{K}, \theta_t, x_t, d(\theta), \xi_0, g_\theta, \partial E_\theta, \lambda, Q$  and the PDP conditions are characterised in terms of this DCPN:

**K:** The domain  $\mathbf{K}$  for the mode process  $\{\theta_t\}$  can be found from the reachability graph (RG) of the DCPN graph. The nodes in the RG are vectors  $V = (v_1, \dots, v_{|\mathcal{P}|})$ , where  $v_i$  equals the number of tokens in place  $P_i, i = 1, \dots, |\mathcal{P}|$ . The RG is constructed from DCPN

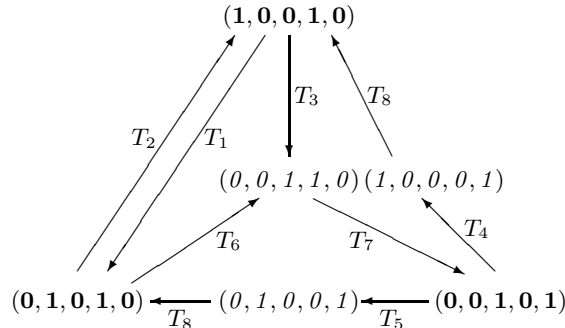
components  $\mathcal{P}$ ,  $\mathcal{T}$ ,  $\mathcal{A}$ ,  $\mathcal{N}$  and  $\mathcal{I}$ . The first node  $V_0$  is found from  $\mathcal{I}$ , which provides the numbers of tokens initially in each of the places<sup>1</sup>. From then on, the RG is constructed as follows: If it is possible to move in one jump from token distribution  $V_0$  to, say, either one of distributions  $V^1, \dots, V^k$  unequal to  $V_0$ , then arcs are drawn from  $V_0$  to (new) nodes  $V^1, \dots, V^k$ . Each of  $V^1, \dots, V^k$  is treated in the same way. Each arc 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 arcs are drawn from  $V^i$  to  $V^j$ . The nodes of the RG become the elements of  $\mathbf{K}$ , with one restriction: Nodes that enable immediate transitions are *excluded* from  $\mathbf{K}$ . To emphasise them, these nodes are given in *italics*. Since the number of places in the DCPN and the number of nodes in the RG are both countable,  $\mathbf{K}$  is a countable set, which satisfies the PDP conditions.

As an example, consider the following DCPN graph, which initially has a token in places  $P_1$  and  $P_4$ , such that  $V_0 = (1, 0, 0, 1, 0)$ . This vector forms the first node of the reachability graph.



Both  $T_1$  and  $T_3$  are pre-enabled. Their guard or delay is ignored. If  $T_1$  fires, it removes the token from  $P_1$  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  $(0, 1, 0, 1, 0)$ . In the reachability graph an arc labelled by  $T_1$  is drawn from  $(1, 0, 0, 1, 0)$  to the new node  $(0, 1, 0, 1, 0)$ . If transition  $T_3$  fires before  $T_1$  does, the token distribution becomes  $(0, 0, 1, 1, 0)$ . Subsequently, the immediate transition  $T_7$  is enabled; its firing leads to  $(0, 0, 1, 0, 1)$ . Since  $(0, 0, 1, 1, 0)$  enables an immediate transition it is drawn in *italics* and is excluded from  $\mathbf{K}$ .

The eventual reachability graph for this example is:



So, for this example,  $\mathbf{K} = \{(1, 0, 0, 1, 0), (0, 1, 0, 1, 0), (0, 0, 1, 0, 1)\}$ .

<sup>1</sup>Notice that  $\mathbf{K}$  can also be constructed independent of  $\mathcal{I}$  by following the proposed procedure for each possible instantiation of the initial token distribution.



- $\theta_t$ : The mode process  $\{\theta_t\}$  is characterised by the vector  $\theta_t = (v_{1,t}, \dots, v_{|\mathcal{P}|,t})$  of length  $|\mathcal{P}|$ , taking values in  $\mathbf{K}$ , where  $v_{i,t}$  denotes the number of tokens in place  $P_i$  at time  $t$ . This characterisation is unique at most time instances, except on those instants at which a transition fires. Uniqueness is constructed as follows: Suppose that  $\tau$  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  $\{T_1, T_2, \dots, T_m\}$  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  $\{\theta_\tau\}$  is defined as that vector  $(v_{1,t}, \dots, v_{|\mathcal{P}|,t}) \in \mathbf{K}$  that occurs after  $T_m$  has fired. This construction also ensures that the process  $\{\theta_t\}$  has limits from the left and is continuous from the right, such that it satisfies the càdlàg property.
- $x_t$ : If  $\theta_t = (v_{1,t}, \dots, v_{|\mathcal{P}|,t}) \in \mathbf{K}$  is the distribution of the tokens among the places of the DCPN at time  $t$ , then  $x_t$  is a vector containing the colours of these tokens. Within a place the colours of the tokens are ordered according to some given algorithm (e.g. according to the time at which they entered the place). Since  $\{\theta_t\}$  is uniquely characterised, so is  $\{x_t\}$ . Since  $\{\theta_t\}$  satisfies the càdlàg property,  $\{x_t\}$  does too. An additional case occurs, however, when  $\theta_t$  jumps to the same value again, so that only the process  $x_t$  makes a jump at time  $\tau$ . In that case, let  $x_t$  be defined according to the timing construction as described for  $\theta_t$  above. With this,  $x_t$  satisfies the càdlàg property.
- $d(\theta)$ : The colour of a token in a place  $P$  is an element of  $\mathcal{C}(P) = \mathbb{R}^{n(P)}$ , therefore if  $\theta_t$  assumes value  $\theta = (\theta_1, \dots, \theta_{|\mathcal{P}|}) \in \mathbf{K}$ , then  $x_t \in \mathbb{R}^{d(\theta)}$ , with  $d(\theta) = \sum_{i=1}^{|\mathcal{P}|} \theta_i \times n(P_i)$ .
- $\xi_0 = (\theta_0, x_0)$ : The DCPN initial marking  $\mathcal{I}$  provides the places the tokens are initially in, from which the initial mode  $\theta_0$  follows (see  $\theta_t$ ), and the colours of these tokens, from which the initial continuous state  $x_0$  follows (see  $x_t$ ).
- $g_\theta$ : The colour  $x_{i,j,t}$  of the  $j$ th token in place  $P_i$  at time  $t$  is determined by the ordinary differential equation  $\dot{x}_{i,j,t} = \mathcal{V}_{P_i}(x_{i,j,t})$ . So, if  $x_t$  denotes the vector of all token values  $x_{i,j,t}$ , where  $j$  runs through  $1, \dots, \theta_i$  and  $i$  runs through  $1, \dots, |\mathcal{P}|$ , then  $g_\theta(x_t)$  can be characterised by the corresponding vector with elements  $\mathcal{V}_{P_i}(x_{i,j,t})$ . Since, for all  $P_i$ ,  $\mathcal{V}_{P_i}$  is locally Lipschitz continuous,  $g_\theta$  is also locally Lipschitz continuous.
- $\partial E_\theta$ : The boundary  $\partial E_\theta$  of subset  $E_\theta$  is determined from the transition guards corresponding with the set of transitions in  $\mathcal{T}_G$  that, under token distribution  $\theta$ , are pre-enabled (this set is uniquely determined). Without loss of generality, suppose this set of transitions is  $T_1, \dots, T_m$ . A boundary is hit when any transition of this set becomes enabled. By DCPN definition, transition  $T_i$  becomes enabled if there is a vector of colours of input tokens  $c_t^i$  that enters boundary  $\partial G_{T_i}$ . This yields  $\partial E_\theta = \partial G'_{T_1} \cup \dots \cup \partial G'_{T_m}$ , where  $G'_{T_i} = [G_{T_i} \times \mathbb{R}^{d(\theta) - |c_t^i|}] \in \mathbb{R}^{d(\theta)}$ , where  $|c_t^i|$  denotes the number of elements in  $c_t^i$  and  $[\cdot]$



denotes that all vector components are ordered corresponding with the positions of the tokens in vector  $x_t$ .

$\lambda$ : The jump rate  $\lambda(\theta, \cdot)$  is equal to the sum of the functions  $\delta_T(\cdot)$  of the set of transitions  $T \in \mathcal{T}_D$  that, under token distribution  $\theta$ , are pre-enabled (this set is uniquely determined). 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  $\delta_T$  is integrable for all  $T \in \mathcal{T}_D$ ,  $\lambda$  is also integrable.

$Q$ : 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 and the set  $\mathcal{F}$ . This characterisation is given in Appendix B.

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

$C_2$ : This condition ( $\lambda$  is integrable) follows from the fact that  $\delta_T$  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 ( $EN_t < \infty$ ) follows from assumption  $D_3$ .

This shows that for any DCPN satisfying conditions  $D_1 - D_3$ , we are able to construct a PDP instantiation representing it, which completes the proof of Theorem 2.

## 8 Example PDP

This section presents a PDP that describes the same process as modelled by a DCPN in Section 5, i.e. the path of an aircraft influenced by its engine and its navigation system.

For this example, the PDP mode process  $\{\theta_t\}$  has three components:  $\theta_t = (\theta_t^1, \theta_t^2, \theta_t^3)'$ , where:

$\theta_t^1$  is the Engine system mode, taking values in  $\{Working, Not\ working\}$ .

$\theta_t^2$  is the Navigation system mode, taking values in  $\{Working, Not\ working\}$ .

$\theta_t^3$  is the Aircraft mission mode, taking values in  $\{Not\ completed, Completed\}$ .

This yields that the set  $\mathbf{K}$  has  $2^3 = 8$  elements  $m_1, \dots, m_8$  with:

$m_1 = (Working, Working, Not\ completed)'$	$m_5 = (Working, Working, Completed)'$
$m_2 = (Not\ working, Working, Not\ completed)'$	$m_6 = (Not\ working, Working, Completed)'$
$m_3 = (Not\ working, Not\ working, Not\ completed)'$	$m_7 = (Not\ working, Not\ working, Completed)'$
$m_4 = (Working, Not\ working, Not\ completed)'$	$m_8 = (Working, Not\ working, Completed)'$

The initial mode equals  $\theta_0 = m_1$ . For  $\theta \in \{m_1, m_2, m_3, m_4\}$ ,  $\partial E_\theta = \partial G \times \mathbb{R}^3$ , while for  $\theta \in \{m_5, m_6, m_7, m_8\}$ ,  $E_\theta$  equals  $\mathbb{R}^6$ . The piecewise continuous process part  $\{z_t\}$  has two components:  $z_t = (x_t, v_t)'$ , with  $x_t$  the position and  $v_t$  the velocity of the aircraft. The first table below gives, for each  $\theta \in \mathbf{K}$ , the locally Lipschitz continuous function  $g_\theta(\cdot)$  and the jump rates  $\lambda$  of the Poisson point process. In the second table below,  $Q(\zeta; \xi) = p$  denotes that if  $\xi$  is the value





of the PDP before the hybrid jump, then, with probability  $p$ ,  $\zeta$  is the value of the PDP immediately after the jump.

$\theta$	$g_\theta(\cdot)$	$\lambda$
$m_1$	$\mathcal{V}_1(\cdot)$	$\delta_4 + \delta_6$
$m_2$	$\mathcal{V}_2(\cdot)$	$\delta_3 + \delta_6$
$m_3$	$\mathcal{V}_2(\cdot)$	$\delta_3 + \delta_5$
$m_4$	$\mathcal{V}_2(\cdot)$	$\delta_4 + \delta_5$
$m_5$	0	$\delta_4 + \delta_6$
$m_6$	0	$\delta_3 + \delta_6$
$m_7$	0	$\delta_3 + \delta_5$
$m_8$	0	$\delta_4 + \delta_5$

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 $z$ , $Q(m_7, z; m_8, z) = \frac{\delta_4}{\delta_4 + \delta_5}$ , $Q(m_5, z; m_8, z) = \frac{\delta_5}{\delta_4 + \delta_5}$ .

It may be clear that the PDP is less simple to comprehend and verify than the DCPN representing the same system in Section 5.

## 9 Conclusions

Piecewise Deterministic Markov Processes (PDPs) can be used to describe virtually all complex continuous-time stochastic processes not involving diffusions. However, for complex practical problems it is often difficult to develop a PDP model. This paper has shown that any PDP with a finite discrete state domain can be represented by a newly developed type of Petri Net, named Dynamically Coloured Petri Net (DCPN). Moreover, it has shown that under some conditions each DCPN can be represented by a PDP. A corollary of both results is that there exist into-mappings between PDP and DCPN. The development of a DCPN model for practical situations is similarly simple as it is for a Coloured Petri Net (Jensen, 1992).

The key result of this paper is that this is the first time that proof of the existence of into-mappings between PDPs and Petri Nets has been established. For other Petri Nets such mappings have never been studied, and are often even unfeasible.

## 10 References

1. Alur, R., Courcoubetis, C., Henzinger, T., Ho, P-H. (1993) Hybrid Automata: An algorithmic approach to the specification and verification of hybrid systems, *Hybrid Systems I, Lecture notes in computer science*, pp. 209-229, Springer-Verlag.
2. Bakker, G.J., Blom, H.A.P. (1993) Air Traffic Collision risk modelling, *Proceedings of the 32nd IEEE Conference on Decision and Control*, pp. 1464-1469, NLR report TP 93292 L.

3. Blom, H.A.P., Bakker, G.J., Blanker, P.J.G., Daams, J., Everdij, M.H.C., Klompstra, M.B. (1998) Accident risk assessment for advanced ATM, *2<sup>nd</sup> USA/Europe Air Traffic Management R&D Seminar, Orlando, NLR report TP 99015*.
4. Branicky, M.S. (1995) Studies in Hybrid Systems: Modelling, analysis and control, *PhD thesis, M.I.T., Cambridge, MA*.
5. David, R., Alla, H. (1987) Continuous Petri Nets, *8th European workshop on application and theory of Petri Nets, Saragosse*, pp. 275-294.
6. David, R., Alla, H. (1994) Petri Nets for the modeling of dynamic systems — A survey, *Automatica*, Vol. 30, No. 2, pp. 175-202.
7. Davis, M.H.A. (1984) Piecewise Deterministic Markov Processes: a general class of non-diffusion stochastic models, *Journal Royal Statistical Soc. (B)*, Vol. 46, pp. 353-388.
8. Davis, M.H.A. (1993) Markov models and optimization, Chapman & Hall.
9. Demongodin, I., Koussoulas, N.T. (1998) Differential Petri Nets: Representing continuous systems in a discrete-event world, *IEEE Transactions on Automatic Control*, Vol. 43, No. 4.
10. Everdij, M.H.C., Klompstra, M.B., Blom, H.A.P. (1996) MUFTIS WP3.2, Final report on safety model, Part II: Development of mathematical techniques for ATM safety analysis, *NLR report TR 96197 L*.
11. Everdij, M.H.C., Blom, H.A.P., Klompstra, M.B. (1997) Dynamically Coloured Petri Nets for Air Traffic Management safety purposes, *Proceedings 8th IFAC Symposium on Transportation Systems, Chania, Greece*, pp. 184-189, *NLR report TP 97493 L*.
12. Giua, A., Usai, E. (1996) High-level Hybrid Petri Nets: a definition, *Proceedings 35th Conference on Decision and Control, Kobe, Japan*, pp. 148-150.
13. Jensen, K. (1992) Coloured Petri Nets: Basic concepts, analysis methods and practical use, Volume 1, Springer-Verlag.
14. Le Bail, J., Alla, H., David, R. (1991) Hybrid Petri Nets, *European Control Conference, Grenoble, France*, pp. 1472-1477.
15. Lygeros, J., Pappas, G.J., Sastry, S. (1998), An approach to the verification of the Center-TRACON automation system, *Proceedings 1st International Workshop Hybrid Systems: Computation and Control*, pp. 289-304.
16. Protter, P. (1990), Stochastic integration and differential equations, a new approach, Springer-Verlag.
17. van Schuppen, J.H. (1998) A sufficient condition for controllability of a class of hybrid systems, *Proceedings 1st International Workshop Hybrid Systems: Computation and Control*, pp. 374-383.
18. Sipser, M. (1997) Introduction to the theory of computation, PWS publishing company, Boston.



19. Tomlin, C., Lygeros, J., Sastry, S (1998) Synthesising controllers for nonlinear hybrid systems, *Proceedings 1st International Workshop Hybrid Systems: Computation and Control*, pp. 360-373.
20. Trivedi, K.S., Kulkarni, V.G. (1993) FSPNs: Fluid Stochastic Petri Nets, *Lecture notes in Computer Science*, Vol. 691, M. Ajmone Marsan (ed.) *Proceedings 14th International Conference on Applications and theory of Petri Nets*, pp. 24-31, Springer Verlag, Heidelberg.
21. Yang, Y.Y., Linkens, D.A., Banks, S.P. (1995) Modelling of hybrid systems based on Extended Coloured Petri Nets, *Hybrid Systems II*, P. Antsaklis *et al.* (eds.), pp. 509-528, Springer.
22. Weinberg, H.B., Lynch, N., Delisle, N. (1996) Verification of automated vehicle protection systems, *Hybrid Systems III, Verification and control*, R. Alur *et al.* (eds.) pp. 101-113, Springer.

## Appendices

### A Formal definition of Dynamically Coloured Petri Nets

This appendix presents a formal definition of Dynamically Coloured Petri Net. As much as possible, the notation introduced by Jensen (1992) for Coloured Petri Net is used.

#### Definition:

A Dynamically Coloured Petri Net (DCPN) is an 11-tuple  $DCPN = (\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{C}, \mathcal{V}, \mathcal{G}, \mathcal{D}, \mathcal{F}, \mathcal{I})$ , together with some rules. Below, first the structure of the components in the tuple is given, next the DCPN evolution through time is explained.

#### Structure of DCPN elements:

1.  $\mathcal{P}$  is a finite set of places.
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  $\mathcal{T}_G \cap \mathcal{T}_D = \mathcal{T}_D \cap \mathcal{T}_I = \mathcal{T}_I \cap \mathcal{T}_G = \emptyset$ .
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  $\mathcal{A}_O \cap \mathcal{A}_E = \mathcal{A}_E \cap \mathcal{A}_I = \mathcal{A}_I \cap \mathcal{A}_O = \emptyset$ .
4.  $\mathcal{N} : \mathcal{A} \rightarrow \mathcal{P} \times \mathcal{T} \cup \mathcal{T} \times \mathcal{P}$  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.



6.  $\mathcal{C} : \mathcal{P} \rightarrow \mathcal{S}$  is a colour function which maps each place  $P \in \mathcal{P}$  to a specific colour type in  $\mathcal{S}$ . Further notation:  $\mathcal{C}(\{P_1, P_2\})$  is short for  $\mathcal{C}(P_1) \times \mathcal{C}(P_2)$ .
7.  $\mathcal{I} : \mathcal{P} \rightarrow \mathcal{C}(P)_{ms}$  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. The initial marking is such, that no immediate transition is immediately enabled.
8.  $\mathcal{V}$  is set of a token colour functions. For each place  $P \in \mathcal{P}$  it contains a locally Lipschitz continuous function  $\mathcal{V}_P : \mathcal{C}(P) \rightarrow \mathcal{C}(P)$ .
9.  $\mathcal{G}$  is a set of transition guards. For each  $T \in \mathcal{T}_G$ , it contains a transition guard  $\mathcal{G}_T : \mathcal{C}(P(A_{in,OE}(T))) \rightarrow \{\text{True}, \text{False}\}$ .  $\mathcal{G}_T(c_t)$  evaluates to True when  $c_t$  enters  $\partial G_T$  for the first time, where  $G_T$  is an open subset in  $\mathcal{C}(P(A_{in,OE}(T)))$ .
10.  $\mathcal{D}$  is a set of transition delays. For each  $T \in \mathcal{T}_D$ , it contains a transition delay  $\mathcal{D}_T : \mathcal{C}(P(A_{in,OE}(T))) \rightarrow \mathbb{R}_0^+$ , which, if evaluated from stopping time  $\tau$  on, follows  $\mathcal{D}_T(c_t) = \inf\{t \mid 1 - e^{-\int_{\tau}^t \delta_T(c_s) ds} \geq u\}$ , where  $\delta_T : \mathcal{C}(P(A_{in,OE}(T))) \rightarrow \mathbb{R}_0^+$  is integrable and  $u$  is a random number drawn from  $U[0, 1]$  at  $\tau$ .
11.  $\mathcal{F}$  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  $\{0, 1\}^{|A_{out}(T)|} \times \mathcal{C}(P(A_{out}(T)))$ .

**DCPN evolution:**

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  $\tau$ , 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 differential equation  $\dot{c}_t = \mathcal{V}_P(c_t)$  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  $\tau_1 = \inf\{t \mid T \text{ is pre-enabled at time } t\}$ . Consider one token per ordinary and enabling arc in the input places of  $T$  and write  $c_t \in \mathcal{C}(P(A_{in,OE}(T)))$ ,  $t \geq \tau_1$ , 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, in case one input place contains several tokens per arc), all possible such vectors are evaluated in parallel.

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

If  $T$  is enabled, suppose this occurs at time  $\tau_2$ , it removes one token per arc in  $A_{in,O}(T)$  from each of its input places. At this time  $\tau_2$ ,  $T$  produces zero or one token along each output arc: If

$c_{\tau_2}$  is the vector of colours of tokens that enabled  $T$  and  $(f, a_{\tau_2})$  is a sample from  $\mathcal{F}_T(\cdot; c_{\tau_2})$ , 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  $a_{\tau_2}$  specifies the colours of the produced tokens. The colours of the new tokens have sample paths that start at time  $\tau_2$ .

In case of ambiguities, the following rules are used:

- $R_0$  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.

**Remark:**

The firing of any transition “stops the time”, i.e. if a transition becomes enabled at time  $\tau$ , it also fires tokens in its output places at time  $\tau$ . And if, for example, there is a sequence of immediate transitions that enable each other with their firing, then their consecutive firings all take place at this same time instant  $\tau$ .

With this, the DCPN definition ends.



## B Characterisation of $Q$ in terms of DCPN

In this appendix,  $Q$  is characterised in terms of DCPN, as part of the characterisation in Section 7 of PDP in terms of DCPN.

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 PDP just prior to the jump is  $(\theta, x)$ , then the value of the PDP just after the jump is in  $(\theta', A)$ . Measure  $Q(\theta', A; \theta, x)$  is characterised in terms of the DCPN by the reachability graph (RG) (see Section 7) and the set  $\mathcal{F}$ , as below.

In the RG, consider nodes  $\theta$  and  $\theta'$  and delete all other nodes that are elements of  $\mathbf{K}$ , including the arcs attached to them. Also, delete all nodes and arcs that are not part of a directed path from  $\theta$  to  $\theta'$ . The residue is named reduced RG (RRG). Then,

- $Q(\theta', A; \theta, x) = 0$  if  $\theta$  and  $\theta'$  are not connected in the RRG by at least one path.

If  $\theta$  and  $\theta'$  are connected then in the RRG one or more paths from  $\theta$  to  $\theta'$  can be identified. Each such path may consist of only one arc, or of sequences of directed arcs that pass nodes that enable immediate transitions. All arcs are labelled by names of transitions, therefore the paths between  $\theta$  and  $\theta'$  may be characterised by the labels on these arcs, i.e. by the transitions that consecutively fire in the jump from  $\theta$  to  $\theta'$ . Denote the set of paths, characterised by these labels, by  $\mathcal{L}$ .

Examples of elements of  $\mathcal{L}$  are  $T_1$  (if  $T_1$  is pre-enabled in  $\theta$  and its firing leads to  $\theta'$ ),  $T_1 + T_2$  (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  $\theta'$ ),  $T_4 \circ T_3$  (if  $T_3$  is pre-enabled in  $\theta$ , its firing leads to the immediate transition  $T_4$  being enabled, and the firing of  $T_4$  leads to  $\theta'$ ), etcetera.

If the set  $\mathcal{L}$  contains only one path, then it is uniquely determined which transition (or set of transitions) will fire in the jump. However, if there are several paths between  $\theta$  and  $\theta'$ , then, even with additional knowledge of  $x$  and  $A$ , the set of transitions that will actually fire is not always uniquely determined. This makes that characterisation of  $Q(\theta', A; \theta, x)$  directly in terms of  $\mathcal{F}_T$  is ambiguous. For this reason, we factorise  $Q$  by conditioning on this set.

Under the condition that a jump occurs:

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

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

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

**Characterisation of  $p_{L|\theta,x}(L | \theta, x)$**

First, assume that  $\mathcal{L}$  does not contain immediate transitions. This yields: each  $L \in \mathcal{L}$  either contains one or more guard transitions, or one delay transition (other combinations occur with zero probability). For each  $T$ , denote by  $c_T^x$  the vector containing those components of  $x$  that correspond with the input tokens of  $T$ . If this vector is not unique, then each such possible vector is evaluated in parallel. Then for each  $L \in \mathcal{L}$ ,  $p_{L|\theta,x}(L | \theta, x)$  is determined in two steps.

1. Consider the subset of paths in  $\mathcal{L}$  that contain only guard transitions (this subset is denoted by  $\mathcal{L} \cap \mathcal{T}_G$ , although this denotation is not entirely correct). For each  $L$  in this subset, define the auxiliary variable  $\alpha_L$  by  $\alpha_L = 1$  if  $c_T^x \in \partial G_T$  for all  $T \in L$ , and  $\alpha_L = 0$  if there is  $T \in L$  for which  $c_T^x \notin \partial G_T$ . Next, if there are  $j$  and  $k$  for which  $\alpha_{L_j} = \alpha_{L_k} = 1$  and for which the transitions in  $L_j$  are a subset of the transitions in  $L_k$ , then redefine  $\alpha_{L_j} = 0$ .

$$p_{L|\theta,x}(L | \theta, x) = 0 \text{ if } \sum_{\ell \in \mathcal{L} \cap \mathcal{T}_G} \alpha_\ell = 0 \text{ and}$$

$$p_{L|\theta,x}(L | \theta, x) = \alpha_L / \sum_{\ell \in \mathcal{L} \cap \mathcal{T}_G} \alpha_\ell \text{ otherwise.}$$

2. For each  $L \in \mathcal{L} \cap \mathcal{T}_D$ :  $p_{L|\theta,x}(L | \theta, x) = 0$  if  $\sum_{\ell \in \mathcal{L} \cap \mathcal{T}_G} \alpha_\ell > 0$  and

$$p_{L|\theta,x}(L | \theta, x) = \frac{\delta_L(c_L^x)}{\sum_{T \in \mathcal{L} \cap \mathcal{T}_D} \delta_T(c_T^x)} \text{ if } \sum_{\ell \in \mathcal{L} \cap \mathcal{T}_G} \alpha_\ell = 0.$$

Next, consider the situations where the RRG 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

$$p_{L|\theta,x}(L | \theta, x) = p_{T_k|\theta,x}(T_k | \theta, x), \text{ with the right-hand-side constructed as above for the case without immediate transitions. The same value } p_{T_k|\theta,x}(T_k | \theta, x) \text{ follows for cases like } L = T_m \circ T_j \circ T_k, \text{ with } T_j \text{ and } T_m \text{ immediate transitions.}$$

However, if the firing of  $T_k$  enables more than one immediate transition, then the value of  $p_{T_k|\theta,x}(T_k | \theta, x)$  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

$$p_{L_1|\theta,x}(L_1 | \theta, x) = p_{L_2|\theta,x}(L_2 | \theta, x) = \frac{1}{2} p_{T_k|\theta,x}(T_k | \theta, x).$$

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

**Characterisation of  $p_{\theta',x'|\theta,x,L}(\theta', A | \theta, x, L)$**

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





- $p_{\theta', x' | \theta, x, L}(\theta', A | \theta, x, L) = 0$  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  $\theta$  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 DCPN state. This gives:

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

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

- If  $L$  consists of several transitions  $T_1, \dots, T_m$  that, given  $\theta$  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  $T_1, \dots, T_m$ :

$$p_{\theta', x' | \theta, x, L}(\theta', A | \theta, x, L) = \sum_{c_{T_1}^x, \dots, c_{T_k}^x} \int_{(e'_1, c'_1), \dots, (e'_k, c'_k)} \mathcal{F}_{T_1}(e'_1, c'_1; c_{T_1}^x) \times \dots \times \mathcal{F}_{T_k}(e'_k, c'_k; c_{T_k}^x) \times \mathbf{I}_{(\theta', A; e'_1, c'_1, c_{T_1}^x, \dots, e'_k, c'_k, c_{T_k}^x)},$$

where  $\mathbf{I}_{(\theta', A; e'_1, c'_1, c_{T_1}^x, \dots, e'_k, c'_k, c_{T_k}^x)}$  denotes indicator function for the event that the combined removal of  $c_{T_1}^x$  through  $c_{T_k}^x$  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 DCPN state in  $(\theta', A)$ .

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

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

where  $\mathbf{I}_{(\theta', A; e'_j, c'_j, e'_k, c'_k, c_{T_k}^x)}$  denotes indicator function for the event that the removal of  $c_{T_k}^x$  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 DCPN state in  $(\theta', A)$ .

- 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 construction,  $Q$  is uniquely characterised in terms of DCPN.