DOCUMENT CONTROL SHEET

	ORIGINATOR'S REF. NLR TP 97493 U		SECURITY CLASS. Unclassified

ORIGINATOR
National Aerospace Laboratory NLR, Amsterdam, The Netherlands

TITLE
Dynamically Coloured Petri Nets for Air Traffic Management Safety Purposes

PRESENTED AT
The 8th IFAC Symposium on Transportation Systems, June 16-18, 1997, Chania, Greece

AUTHORS	DATE	pp	ref
M.H.C. Everdij, H.A.P. Blom M.B. Klompstra	971008	10	9

DESCRIPTORS

Aircraft safety	Markov processes
Air traffic control	Mathematical models
Collision avoidance	Petri nets
Flight paths	Probability theory
Flight safety	Stochastic processes

ABSTRACT
Piecewise Deterministic Markov processes (PDPs) are known to describe in a most general way the evolution of stochastic models which include both discrete and continuous processes. However, the development and evaluation of PDPs is very difficult. This paper proposes an extension of the Petri net concept with which PDPs can be described. The advantage is the modelling power, the analysis ability and the graphical support and visualisability. The Petri net extension is next illustrated with an Air Traffic Management example, in which aircraft consecutively make missed approaches on two converging runways.

NLR TECHNICAL PUBLICATION

TP 97493 U

DYNAMICALLY COLOURED PETRI NETS FOR AIR TRAFFIC MANAGEMENT SAFETY PURPOSES

by

M.H.C. Everdij

H.A.P. Blom

M.B. Klompstra

This paper was presented at the 8th IFAC Symposium on Transportation Systems, June 16-18, 1997, Chania, Greece

Division	: Flight	Completed	: 971008
Prepared	: MHCE/ HAPB/ MBK/	Order number	: 537405
Approved	: JB/	Typ.	: MHCE

Contents

(10 pages in total)

This page is intentionally left blank

DYNAMICALLY COLOURED PETRI NETS FOR AIR TRAFFIC MANAGEMENT SAFETY PURPOSES

M.H.C. Everdij, H.A.P. Blom and M.B. Klompstra

National Aerospace Laboratory NLR
P.O. Box 90502, 1006 BM Amsterdam, The Netherlands
E-mail: everdij@nlr.nl

Abstract. Piecewise Deterministic Markov processes (PDPs) are known to describe in a most general way the evolution of stochastic models which include both discrete and continuous processes. However, the development and evaluation of PDPs is very difficult. This paper proposes an extension of the Petri net concept with which PDPs can be described. The advantage is the modelling power, the analysis ability and the graphical support and visualisability. The Petri net extension is next illustrated with an Air Traffic Management example, in which aircraft consecutively make missed approaches on two converging runways.

Keywords. Markov models, Hybrid modes, Petri nets, Air Traffic Control, Safety.

1. INTRODUCTION

The motivation for this research is to develop a framework for modelling and evaluating safety of advanced Air Traffic Management (ATM) procedures, through stochastic models which include both discrete and continuous processes. Davis (1984, 1993) has shown that processes of this kind can most generally be described by Piecewise Deterministic Markov Processes (PDPs). The development and evaluation of a PDP model for a complex application is very difficult. In this paper, Dynamically Coloured Petri Nets are introduced to represent such PDPs. This Petri net extension supplies insight into the complex mathematical structure PDPs form, and it may support modelling complex systems using PDPs. This support is necessary because it is easy to lose track of the system elements and their relationships if PDPs are only used at the strict mathematical level.

In this paper, first Piecewise Deterministic Markov Processes and Petri nets are described shortly and the current status of Petri net extensions in existing literature is clarified. Next, a new Petri net extension which covers PDPs is introduced. Finally, this extension is illustrated with an ATM example. This example makes clear that the set of processes which can be modelled using Petri nets is highly enlarged by the introduction of this extension and that it is yet possible to have graphical support when modelling with Piecewise Deterministic Markov Processes. This support will definitely lower the threshold for people who want to model and simulate complex dynamic processes, but who are afraid of PDP mathematics.

Part of the research is supported by DGVII of the European Union (Everdij *et al.* 1996a) and Netherlands Department of Civil Aviation (RLD) (Everdij *et al.* 1996b).

2. PIECEWISE DETERMINISTIC MARKOV PROCESS

In the words of Davis (1993), Piecewise Deterministic Markov Processes (PDPs) 'cover virtually all continuous-time stochastic models arising in applications, except diffusions'. A PDP consists of two components: a discrete valued component $\{\theta_t\}$ and a continuous valued component $\{x_t\}$. The discrete valued component models the current mode of the Markov process. At discrete times, it may switch to another mode which is selected according to some probabilistic relation. The continuous valued component models the dynamic evolution of the Markov process. At discrete moments in time, it may jump according to some relation, which makes it only piecewise continuous. The PDP state is given by the states of these two components together, $\xi_t = \mathrm{Col}\{\theta_t, x_t\}$, and is called a *hybrid state*. The times at which the switches and jumps occur depend on a doubly stochastic Poisson process generating a point, or are triggered if the piecewise continuous valued process hits the boundary of a predefined area. A switch and a jump may either occur at the same time, or at different times. If $\{x_t\}$ also makes a jump at times when $\{\theta_t\}$ switches, this is called a hybrid jump.

For each θ in its domain $\mathbf{K}$, let E_θ be an open subset of $\mathbb{R}^n$ and define E by $E \triangleq \{\mathrm{Col}\{\theta, x\} | \theta \in \mathrm{K}, x \in E_\theta\}$. For each $\theta \in \mathrm{K}$, $\mathcal{X}_\theta$ is a given locally Lipschitz continuous vector field in E_θ, determining a flow $\phi_{\theta,x}(t)$, with starting point $\phi_{\theta,x}(0) \triangleq x$. Now, suppose at some stopping time τ the process state is given by $\xi_\tau = \xi = \mathrm{Col}\{\theta, x\} = \mathrm{Col}\{\theta, \phi_{\theta,x}(0)\}$. If no jumps occur, the process state at $t \geq \tau$ is given by $\xi_t = \mathrm{Col}\{\theta_t, x_t\} = \mathrm{Col}\{\theta, \phi_{\theta,x}(t-\tau)\}$. At some stopping times, however, the process state may jump. These stopping times are generated by either one of the following events, depending on which generates a point first:

(1) A Poisson process generates a point with rate $\lambda(\theta_t, x_t)$, $t > \tau$.
(2) The process state $\xi_t = \mathrm{Col}\{\theta_t, x_t\}$ hits the boundary ∂E of E, $t > \tau$.

This yields for the survivor function of the first jump, starting from point $\xi = \mathrm{Col}\{\theta, x\}$ 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. The function $t_*(\theta, x)$ denotes the time until the first boundary hitting after $t = \tau$ and is given by $t_*(\theta, x) \triangleq \inf\{t - \tau > 0 | \phi_{\theta,x}(t-\tau) \in \partial E_\theta\}$.

The time until the first jump is calculated by drawing a sample from $G_\xi(\cdot)$. It is assumed that it is also possible to draw a multi-dimensional sample z from Q, the transition measure for the magnitude of the jump given the current value of the hybrid state. With this, the algorithm to determine a sample path for the hybrid state process ξ_t, $t \geq 0$, from the initial state $\xi_0 = \mathrm{Col}\{\theta_0, x_0\}$ is in two steps; define $\tau_0 \triangleq 0$ and let for $k = 0$, $\xi_{\tau_k} = \mathrm{Col}\{\theta_{\tau_k}, x_{\tau_k}\}$ be the initial state, then for $k = 1, 2, \ldots$:

Step 1: Draw a sample σ_k from function $G_{\xi_{\tau_{k-1}}}(\cdot)$, then the time τ_k of the kth jump is $\tau_k = \tau_{k-1} + \sigma_k$. For $\tau_{k-1} \leq t < \tau_k$ and $\tau_k \leq \infty$ the sample path up to the kth jump is then given by

$$\xi_t = \mathrm{Col}\{\theta_{\tau_{k-1}}, \phi_{\theta_{\tau_{k-1}}, x_{\tau_{k-1}}}(t - \tau_{k-1})\}.$$

Step 2: For $\xi'_{\tau_k} = \mathrm{Col}\{\theta_{\tau_{k-1}}, \phi_{\theta_{\tau_{k-1}}, x_{\tau_{k-1}}}(\tau_k - \tau_{k-1})\}$, draw a multi-dimensional sample z_k from transition measure $Q(\cdot; \xi'_{\tau_k})$, then the process state at the time τ_k of the kth jump is given by $\xi_{\tau_k} = z_k$, if $\tau_k < \infty$.

3. COLOURED PETRI NET

An Ordinary Petri Net (Petri, 1962) is a directed graph with two types of nodes: places and transitions. The transitions (rectangles) model actions, the places (circles) represent possible pre or post conditions for these actions. The arcs (arrows) connect places with transitions. A pre condition is said to be current if the corresponding place contains a token (a dot). A transition is enabled if all of the places by which it is connected through an incoming arc (its input places) contain at least one token. If this occurs, the transition fires, it removes one token from each of its input places and produces one token for each of its output places. The current state of the Petri net is called marking and is denoted by a vector counting the current number of tokens in each of the places.

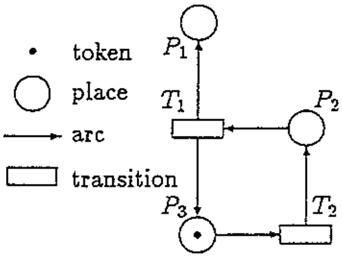
In a more formal representation, an Ordinary Petri Net is a four-tuple $OPN = (\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{M}_0)$, where

(1) $\mathcal{P}$ denotes the set of places.
(2) $\mathcal{T}$ denotes the set of transitions, $\mathcal{T} \cap \mathcal{P} = \emptyset$.
(3) $\mathcal{A} \subseteq \mathcal{P} \times \mathcal{T} \cup \mathcal{T} \times \mathcal{P}$ denotes the set of arcs.
(4) $\mathcal{M}_0 : \mathcal{P} \to \mathbb{N}$ is the initial marking function, which counts the number of tokens initially resident in each of the places.

If a transition T is enabled, it fires, changing its current marking $\mathcal{M}_1$ into marking $\mathcal{M}_2$, through

$$\mathcal{M}_2(P) = \begin{cases} \mathcal{M}_1(P) - 1 & \text{if } (P,T) \in \mathcal{A}, (T,P) \notin \mathcal{A} \\ \mathcal{M}_1(P) + 1 & \text{if } (T,P) \in \mathcal{A}, (P,T) \notin \mathcal{A} \\ \mathcal{M}_1(P) & \text{otherwise} \end{cases}$$

where $\mathcal{M}(P)$ is the number of tokens resident in place P in marking $\mathcal{M}$. A very simple Petri net, both in graphical representation and in the tuple representation is depicted in Figure 1.



$$OPN = (\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{M}_0)$$
$$\mathcal{P} = \{P_1, P_2, P_3\}$$
$$\mathcal{T} = \{T_1, T_2\}$$
$$\mathcal{A} = \{(P_2, T_1), (P_3, T_2) (T_1, P_1), (T_1, P_3), (T_2, P_2)\}$$
$$\mathcal{M}_0 = (0, 0, 1)$$

Fig. 1. A simple Petri net. Transition T_2 is enabled.

Since its introduction, many extensions of Petri nets have been introduced, see e.g. Jensen (1991) and Peterson (1981). However, non of these extensions are sufficient to model Piecewise Deterministic Markov Processes. In particular, the current extensions do not incorporate properly the continuous processes. For this type of processes, another new extension is introduced in the following section. This new model is an extension of the Coloured Petri Net (CPN) model as it was described by Jensen (1991); therefore this CPN model is described in detail next as an example of an existing extension.

A Coloured Petri Net (CPN) is a 9-tuple $CPN = (\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{C}, \mathcal{G}, \mathcal{E}, \mathcal{I})$, where

(1) $\mathcal{P}$ is a finite set of places.
(2) $\mathcal{T}$ is a finite set of transitions, such that $\mathcal{T} \cap \mathcal{P} = \emptyset$.
(3) $\mathcal{A}$ is a finite set of arcs such that $\mathcal{A} \cap \mathcal{P} = \mathcal{A} \cap \mathcal{T} = \emptyset$.
(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 in $\mathcal{A}$ to a pair of nodes.
(5) $\mathcal{S}$ is a finite set of colour types. Generally, these types are products of the usual types, such as string, integer, real, etc.
(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 $\mathcal{C}(P)$. Only tokens of this type may reside in the place.
(7) $\mathcal{G}$ is a guard function. For each $T \in \mathcal{T}$, $\mathcal{G}(T)$ is of type boolean and must evaluate to 'True' before T is allowed to fire.
(8) $\mathcal{E}$ is an arc expression function. For each $A \in \mathcal{A}$, $\mathcal{E}(A)$ defines the number of tokens and the colour of these tokens to be removed (in case of an input arc) or produced (in case of an output arc) at the firing of a transition.
(9) $\mathcal{I}$ is an initialization function which defines the initial marking of the net.

4. DYNAMICALLY COLOURED PETRI NET

The Coloured Petri Net has many advantages compared to the Ordinary Petri Net. Different flows of information can be distinguished by the differently coloured tokens and they can follow different place paths. The CPNs cannot incorporate continuous process flows, however. To solve this, next a new dynamical colouring extension is introduced, in which the colour of a token at some time instant is equal to the solution of a differential equation at that time. The whole concept implies that if this extended Petri net is implemented in a simulator, the set of colours of all tokens and the places in which they reside at some time instant determines the hybrid state of the simulated process. In the sequel this extension is named Dynamically Coloured Petri Net (DCPN), a formal description of which is:

$DCPN$=$(\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{C}, \mathcal{G}, \mathcal{E}, \mathcal{I}, \mathcal{H}, \mathcal{D}, \mathcal{V})$, where $(\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{C}, \mathcal{G}, \mathcal{E}, \mathcal{I})$ is a Coloured Petri Net and

(1) $\mathcal{H}$ is the time horizon on which the simulation of the DCPN is defined. Usually, $\mathcal{H} = [0, T_H]$, where T_H is a positive real. For each time $t \in \mathcal{H}$, $\mathcal{M}(t)$ is the current marking of the Petri net which at the start of the simulation it is equal to the initial marking.
(2) $\mathcal{D}$ is a delay function defined on $\mathcal{T}$. It uses the current marking $\mathcal{M}(t)$ as input parameter. The delay function determines the transition time of each transition which is the time distance between the moment the transition is enabled and the time it produces a token. Notice that during that period the token remains in the input place and might be removed by other transitions.
(3) $\mathcal{V}$ is a token colour function defined on $\mathcal{P}$. It describes what happens to the value of a token while it is residing in a place $P \in \mathcal{P}$. In general the value of the token satisfies a differential equation, which may use the current marking as input parameter.

The transition rules are the same as those for CPN, but one should notice that since the colour of a token may change while it is residing in its place, the guard evaluation is continuously updated.

5. PDP INTO DCPN

In the DCPN modelling the PDP, the current mode of the hybrid state process is modelled by the occurrence of a token in a place labelled with the mode value. The colour of the token is equal to the value of the hybrid state process. A mode switch occurring is modelled by the token being removed from a place and produced into another place (after an intermediate step in an auxiliary place). The evolution of the continuous component of the hybrid state process between jumps is modelled by the flow $\phi_{\theta,x}(t)$. In the DCPN this evolution is defined by the token value function $\mathcal{V}$. A (hybrid) jump can

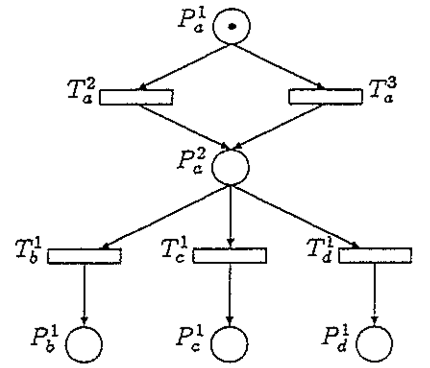


Fig. 2. Petri net modelling a Piecewise Deterministic Markov Process.

occur at random times, either according to some Poisson process or after a boundary hitting occurred.

Figure 2 presents the DCPN modelling a PDP in its graphical representation. For simplicity, the figure is limited to the representation of one hybrid jump and the space for the mode process is supposed to be equal to $\mathrm{K} = \{a, b, c, d\}$. Suppose that at stopping time τ_{k-1} the mode component has value $\theta_{\tau_{k-1}} = a$. In the figure this is modelled by a coloured token residing in place P_a^1. Initially, this token has colour (value) $\xi_{\tau_{k-1}} = \mathrm{Col}\{a, x_{\tau_{k-1}}\}$. The sample path of $\xi_t = \mathrm{Col}\{\theta_t, x_t\}$ up to and at the time of the next jump is determined in two steps:

Step 1: While the token is residing in place P_a^1, the token colour's second component is constantly updated according to the flow $x_t = \phi_{a,x_{\tau_{k-1}}}(t - \tau_{k-1})$. Transition T_a^2 and transition T_a^3 compete for this token which resides in their common input place P_a^1. Transition T_a^2 models the Poisson process generating a mode switch, while transition T_a^3 models the boundary hitting generating a mode switch. The transition which fires first determines the kind of switch occurring. The time at which this is bound to happen is denoted by τ_k-. At this time, the other transition is disabled and will stop its actions. When this occurs, the token is eaten from place P_a^1 by the active transition and a new token is generated for place P_a^2. The index a in the name of this place P_a^2 denotes that the mode value is still equal to $\theta_{\tau_k-} = a$ (a mode switch has not taken place yet). The colour (value) of this new token in place P_a^2 equals the colour of the token which was removed from place P_a^1 at time τ_k-: $\xi_{\tau_k-} = \mathrm{Col}\{a, \phi_{a,x_{\tau_{k-1}}}(\tau_k - \tau_{k-1})\}$.

Step 2: Now, with a token in place P_a^2, transitions T_b^1, T_c^1 and T_d^1 have a token in their common input place P_a^2. Only one of them can fire, so they have to compete for this token. While this token is in P_a^2, its colour is updated for the time instant between τ_k- and τ_k: the size of the jump in the discrete and in the continuous component is computed using transition measure Q. Transition T_b^1 is allowed to fire if the first component of the updated colour appears to be b, and analogously for T_c^1 and T_d^1. The transition which wins, fires a token with the same colour as the one it removes from P_a^2.

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

In the tuple representation, the DCPN modelling the PDP is as follows: suppose the complete system modelled by the PDP is divided into r subsystems or modules. In principle the modules run independently but they can influence each other's behaviour at some points. Suppose that for each of these subsystems a set K_i, $i \in \{1, \ldots, r\}$ of mode values is identified. These modes represent the discrete component values of the subsystem state. The multiset containing the mode values of each K_i is named K. Now, for each $i \in \{1, \ldots, r\}$, the DCPN submodule is represented by:

$DCPN_i = (\mathcal{P}_i, \mathcal{T}_i, \mathcal{A}_i, \mathcal{N}_i, \mathcal{S}_i, \mathcal{C}_i, \mathcal{G}_i, \mathcal{E}_i, \mathcal{I}_i, \mathcal{H}_i, \mathcal{D}_i, \mathcal{V}_i)$, where

(1) $\mathcal{P}_i$ is a finite set of places. For each $\theta \in \mathrm{K}_i$, two places P_θ^1 and P_θ^2 are defined in $\mathcal{P}_i$.
(2) $\mathcal{T}_i$ is a finite set of transitions. This set contains for each element $\theta \in \mathrm{K}_i$ three elements T_θ^1, T_θ^2 and T_θ^3.
(3) $\mathcal{A}_i$ is a finite set of arcs. If K_i contains $|\mathrm{K}_i|$ elements, $\mathcal{A}_i$ contains $2|\mathrm{K}_i|(|\mathrm{K}_i| + 1)$ elements.
(4) $\mathcal{N}_i : \mathcal{A}_i \to \mathcal{P}_i \times \mathcal{T}_i \cup \mathcal{T}_i \times \mathcal{P}_i$ is a node function which maps each arc in $\mathcal{A}_i$ to a pair of nodes. For each element $\theta \in \mathrm{K}_i$, these pairs are (P_θ^1, T_θ^2), (P_θ^1, T_θ^3), (T_θ^2, P_θ^2), (T_θ^3, P_θ^2), and additionally for each ordered pair (θ, ϑ), with $\theta, \vartheta \in \mathrm{K}_i$, $\vartheta \neq \theta$, there are two pairs $(P_\theta^2, T_\vartheta^1)$ and $(T_\vartheta^1, P_\vartheta^1)$.
(5) $\mathcal{S}_i$ is a finite set of colour types. This set only contains one element: the type of the hybrid process ξ_t on this submodule, which is $\mathrm{K}_i \times \mathbb{R}^n$.
(6) $\mathcal{C}_i : \mathcal{P}_i \to \mathcal{S}_i$ is a colour function which maps each place $P \in \mathcal{P}_i$ to a specific colour type $\mathcal{C}_i(P)$. Only tokens of this type may reside in place P. Since only one type of tokens is defined, $\mathcal{C}_i$ has value $\mathcal{C}_i(P) = \mathrm{K}_i \times \mathbb{R}^n$ for all P.
(7) $\mathcal{G}_i : \mathcal{T}_i \to \{\text{True,False}\}$ is a guard function. $\mathcal{G}_i(T)$ must evaluate to 'True' before T can fire. Since the colour of the token in the input place is constantly changing, the guard values have to be constantly evaluated. Three types of transitions are defined in the $DCPN_i$, labelled by 1, 2 or 3. Their guard functions at time t are defined as follows:
$\mathcal{G}_i(T_\theta^1) \equiv \pi_1(\text{token}) = \theta$
$\mathcal{G}_i(T_\theta^2) \equiv -\int_{\tau_{k-1}}^t \lambda_i(\theta, x_s) ds \leq \ln(1 - u_{k-1})$
$\mathcal{G}_i(T_\theta^3) \equiv x_t \in \partial E_{i,\theta}$
Here, $\pi_1(\text{token})$ is the projection on the first com-

ponent of the token colour, λ_i is the rate with which the Poisson process generates mode switches, u_{k-1} is a random number, which is sampled from a uniform distribution on $[0,1]$ and drawn at time τ_{k-1}. $\partial E_{i,\theta}$ is the boundary of an open subset of $\mathbb{R}^n$ such that if the continuous process component enters it, a boundary hitting occurs, which may trigger a mode switch or even a hybrid jump.

(8) $\mathcal{E}_i$ is an arc expression function. It defines what number of tokens of which colour are to be removed (in the case of an input arc to a transition) or produced (in the case of an output arc from a transition) at the firing of a transition. In this PDP model the number of tokens to be removed or produced always equals unity.

(9) $\mathcal{I}_i$ is an initialization function. If the initial hybrid state of the PDP subsystem is $\xi_0 = \mathrm{Col}\{\theta_0, x_0\}$, then all places in the $DCPN_i$ initially contain zero tokens, except for place $P^1_{\theta_0}$, which contains one token of value ξ_0.

(10) $\mathcal{H}_i$ is the time horizon on which the simulation of the $DCPN_i$ is defined, e.g. $\mathcal{H}_i = [0, T_H]$, where T_H is a positive real. For each time $t \in \mathcal{H}_i$, $\mathcal{M}_i(t)$ is the current marking of the Petri net. Since there is only one token in the Petri net submodule, and since with probability one, the colour of this token determines the place in which it is residing, the current marking can be denoted by the current colour of this token, which is ξ_t.

(11) $\mathcal{D}_i$ is a delay function. The delay is zero for all transitions defined in this PDP model.

(12) $\mathcal{V}_i$ is a token colour function. The function describes what happens to the value of a token while it is residing in a place $P \in \mathcal{P}_i$. For places of type P^1_θ, the function $\mathcal{V}_i$ is as follows: If $\xi_{\tau_{k-1}}$ denotes the colour of the token as it enters the place P^1_θ at time τ_{k-1}, then $\mathcal{V}_i(P^1_\theta)$ satisfies $\xi_t = \mathcal{V}_i(P^1_\theta)[\xi_t, \xi_{\tau_{k-1}}] = \mathrm{Col}\{\theta, \phi_{\theta, x_{\tau_{k-1}}}(t-\tau_{k-1})\}$. For places of type P^2_θ, $\mathcal{V}_i$ is defined as follows: If $\xi_{\tau_k -} = \mathrm{Col}\{\theta, \phi_{\theta, x_{\tau_{k-1}}}(\tau_k - \tau_{k-1})\}$ denotes the colour of the token as it enters the place P^2_θ at time τ_k-, then $\mathcal{V}_i(P^2_\theta)$ equals z_k, which is a multi-dimensional sample from transition measure $Q_i(\cdot;\xi_{\tau_k -})$.

6. APPLICATION TO AN ATM EXAMPLE

In the Air Traffic Management example considered, aircraft consecutively make final approaches on two converging runways using a tool named Converging Runway Display Aid (CRDA), (Mundra *et al.*, 1988). During the procedure the humans involved (pilots and air traffic controllers) perform several tasks, during which they may make mistakes. Also the equipment may fail

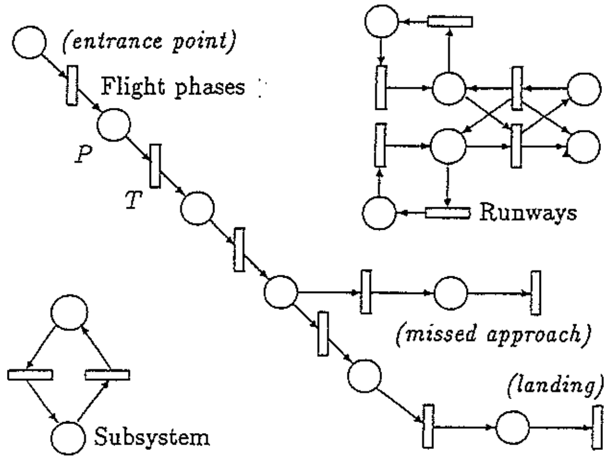


Fig. 3. Dynamically Coloured Petri net for the converging runway example

during some time periods. These failures and external influences, such as wind, all may have some effect on the flightpath, or may even trigger a missed approach. If two aircraft approaching on two converging runways both make a missed approach, they might meet at the point of intersection of the two extended runways, which can be a dangerous situation. The frequency of occurrence of these and other types of events should be measured by simulation of an appropriate Petri net. In the sequel of this section a Dynamically Coloured Petri Net is shortly described for this example. It was developed in Everdij *et al* (1996b) and implemented in NLR tool TOPAZ.

The DCPN developed consists of 7 modules, and additionally for each aircraft in the system 5 modules. Figure 3 presents part of the DCPN in its graphical representation. One of the aircraft dependent modules describes the different flight phases of an aircraft on approach for one of the runways, all other modules describe the nominal, hazardous and failure modes of the different ATM subsystems. The subsystems identified are: pilot, air traffic controller, communication equipment (airborne and ground), runway, weather, aircraft equipment, navigation, air traffic control display, air traffic control system. For each subsystem one nominal mode and (in this example, only) one non-nominal mode is identified, for the runways module two non-nominal modes are identified: one (for each runway) local failure mode and one failure mode common for both runways. In the DCPN each mode is identified by a place. The simple subsystems have two places and two transitions which establish switches from one mode to the other. Research was done to identify appropriate transition rates, which are all assumed Poisson.

For the flight phases module the flight of an aircraft is divided into phases, such as until outer marker, outer marker till protection boundary, protection boundary till treshold, treshold till touchdown, landing. These phases are linked into a string of places and transitions. A token in this module represents an aircraft of which the position and velocity follow the solution of differential equations which have a form and parameters typical for that place/phase it is in. If the aircraft crosses a boundary of a flight segment (e.g. passes the outer marker), the associated transition (in this case the output transition of place till outer marker) is enabled and moves the token to the next place in the string. The parameters of the differential equations are influenced by the state of the other subsystem modules, e.g. if there is a strong wind, the velocity of the aircraft will change. In some combinations of modes, it might even happen that a safe landing of the aircraft is expected to be compromised. In that case the aircraft may make a missed approach. The token colour function for place P and the guard for transition T as depicted in Figure 3 are given below in more detail as an illustration.

The colour of a token in place P: flight phase 'till outer marker' is a vector $x(t)$ with three position components (longitudinal $//$, lateral $\vdash$ and vertical $\perp$) and three velocity components. It is written as a weighted sum of a reference state $\bar{x}(t)$ and a deviation state $\tilde{x}(t)$, for each of which differential equations are developed:

$$\dot{\bar{x}}(t) = \begin{bmatrix} 0 & 0 & & & & \\ 0 & 0 & & & & \\ & & 0 & 0 & & \\ & & 0 & 0 & & \\ 0 & -\tan\alpha & & & 0 & 0 \\ 0 & 0 & & & 0 & 0 \end{bmatrix} \bar{x}(t) + \begin{bmatrix} v_{OM} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where α is glideslope angle.
Actual state; nominal case:

$$x(t) = \bar{x}(t) + \begin{bmatrix} 1 & 0 & & & & \\ 0 & 1 & & & & \\ & & 1 & 0 & & \\ & & 0 & 1 & & \\ \tan\alpha & 0 & & & 1 & 0 \\ 0 & \tan\alpha & & & 0 & 1 \end{bmatrix} \tilde{x}(t), \text{ where}$$

$$d\tilde{x}(t) = \begin{bmatrix} 0 & 1 & & & & \\ a_1 & a_2 & & & & \\ & & a_3(t) & 0 & & \\ & & 0 & a_4 & & \\ & & & & a_5(t) & 0 \\ & & & & 0 & a_6 \end{bmatrix} \tilde{x}(t)dt,$$

where $a_3(t)$ and $a_5(t)$ are appropriate functions of time which have to be estimated using statistical data. Similar equations are developed for the non-nominal cases.

The guard of transition T, aircraft passes outer marker abeam, is defined by:

$$\mathcal{G}(T) = \begin{cases} \text{True, if } x_{//} \geq -d_{OM} \\ \text{False, if } x_{//} < -d_{OM} \end{cases}$$

where $-d_{OM}$ is the longitudinal component of the position of the outer marker.

7. CONCLUDING REMARKS

This paper has shown that modelling of Piecewise Deterministic Markov Processes (PDPs) can be graphically supported by the use of a newly developed Petri net extension, named Dynamically Coloured Petri Nets (DCPNs). This extension provides insight in the combination of continuous processes with discrete event processes. The DCPN described was implemented as module of NLR's tool TOPAZ (Traffic Organization and Perturbation AnalyZer). Sequences of aircraft on approach were simulated with the aim to count frequencies of missed approaches under different combinations of non-nominal modes. The results were used as input for a Generalised Reich Collision Risk model (Bakker and Blom, 1993) in order to determine collision risk at the intersection of the two converging runways. Follow-on research will incorporate diffusion and address the human factor more accurately.

REFERENCES

Bakker, G.J. and H.A.P. Blom (1993). Air Traffic Collision risk modelling. *Proc. 32nd IEEE Conf. on Decision and Control.* pp.1464-1469.

Davis, M.H.A. (1984). Piecewise Deterministic Markov Processes: a general class of non-diffusion stochastic models. *J. Royal Statist. Soc. (B)*, 46, 353-388.

Davis, M.H.A. (1993). *Markov models and optimization.* Chapman & Hall.

Everdij, M.H.C., M.B. Klompstra and H.A.P. Blom (1996a). *MUFTIS WPR 3.2, Final report on safety model Part II: Development of mathematical techniques for ATM safety analysis.* NLR TR 96197 L.

Everdij, M.H.C., G.J. Bakker and H.A.P. Blom (1996b). *Application of Collision Risk Tree analysis to DCIA/CRDA with support of TOPAZ.* NLR CR 96784 L.

Jensen, K. (1991). Coloured Petri Nets: A high level language for system design and analysis. In: *High level Petri Nets, theory and application.* Springer-Verlag. K. Jensen, G. Rozenberg, Eds.). pp.44-119.

Mundra, A.D. *et al* (1988). Simulation test results of a controller's display aid for conducting staggered approaches to converging runways. *34th Annual Meeting of the Air Traffic Control Association.* pp.462-473.

Peterson, J.L. (1981). *Petri net theory and the modeling of systems*, Prentice Hall.

Petri, C.A. (1962). *Kommunikation mit Automaten.* PhD Dissertation (in German). Schriften des Institutes für Instrumentele Mathematik. Bonn, Germany.