Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Stochastic semantics for Communicating Piecewise Deterministic Markov Processes

Stefan Strubbe and Arjan van der Schaft

***Abstract*— CPDPs (Communicating Piecewise Deterministic Markov Processes) can be used for compositional specification of systems from the class of stochastic hybrid processes formed by PDPs (Piecewise Deterministic Markov Processes). We give an extension of the CPDP model of [6]. This extension provides richer interaction possibilities such as broadcasting (and recieving) of multiple signals at the same time. We show that the $|^P_A|$ operator from [7] can be used in the context of CPDPs to express all these interactions. We provide an algorithm in which scheduling and maximal progress are used to find the PDP that models the behavior of a CPDP of this new type. We give the conditions under which a CPDP allows this PDP-semantics.**

I. INTRODUCTION

Many real-life systems nowadays are complex hybrid systems. They consist of multiple components 'running' simultaneously, having both continuous and discrete dynamics and interacting with each other. Also, many of these systems have a stochastic nature. An interesting class of stochastic hybrid systems is formed by the Piecewise Deterministic Markov Processes (PDPs), which were introduced in 1984 by Davis (see [1], [2]). Motivation for considering PDP systems is twofold. First, almost all stochastic hybrid processes that do not include diffusions can be modelled as a PDP, and second, PDP processes have very nice properties (such as the strong Markov property) when it comes to stochastic analysis. (In [2] powerful analysis techniques for PDPs have been developed.) However, PDPs cannot communicate or interact with other PDPs and therefore the PDP-framework does not allow compositional modelling (where all components are modelled individually and connected / composed afterwards).

In [6] the CPDP automata framework is introduced as a compositional modelling framework for PDP-type systems. A CPDP is an open system which can interact with other CPDPs. CPDPs can be connected / composed via a parallel composition operator. A CPDP can be closed (by 'closing down all interaction channels') and in [8] it is proved that the behavior of a closed CPDP can be modelled through a PDP. This means that CPDPs can be used for compositional modelling of complex stochastic hybrid systems and that composite CPDPs (which contain all relevant components and do therefore not interact anymore with other components) can be analyzed by using PDP analysis techniques. Another framework that has been developed for compositional modelling of PDP-type systems is the Petri net framework DCPN (Dynamically Coloured Petri Nets) of [3].

In this paper we upgrade the compositional power of CPDPs in two ways. First, by using guards instead of the boundaries of the CPDPs in [6], we show that it is possible to exploit the full compositional power of the $|^P_A|$ composition operator of [7] in the context of CPDPs. $|^P_A|$ is a rich composition operator, defined in [7] for general transition systems with active and passive transitions, which allows several types of interaction by exploiting all possible combinations of synchronization of active and passive transitions.

Second, by using guards we can allow that multiple transitions are executed (in a specific order) at the same time instant. This means that at one time instant a chain of signals can be broadcast between the components of complex CPDP. This feature has proved to be very useful in for example the Air Traffic Management CPDP model in [10].

By using guards in the CPDP model, we diverge from the PDP model because non-determinism (introduced by the guards) and mutliple-transitions-at-the-same-time-instant are not present in the PDP model. However, we show that under certain conditions, the behavior of CPDPs can still be modelled through PDPs. We do this by presenting an algorithm which replaces hybrid jumps of multiplicity greater than one (i.e. a chain of multiple transitions executed at the same time instant) by a single transition. We show that converting a chain of transitions to a single transition does not change the stochastic behavior. With this algorithm CPDPs can be converted to CPDPs of the old type (as in [6]) which means that its behavior can be modelled through a PDP. We give necessary and sufficient conditions under which this conversion is possible. Then, if conversion is possible, CPDPs of the new type still allow analysis via PDP-analysis-techniques.

The organization of this paper is as follows. In Section II we give the definition of the CPDP automaton and we highlight how this definition differs from the definition of [6]. In Section III composition of CPDPs is defined via the $|^P_A|$ operator. In Section IV we present the algorithm, and the conditions under which it works, that converts a CPDP to a CPDP of the old type. Therefore, this algorithm, which provides a stochastic PDP semantics for CPDP, can be used to find the PDP that models the behavior of a CPDP of the new type. Finally, in Section V we draw conclusions.

II. DEFINITION OF CPDP

We give the formal definition of CPDP as an automaton.

This work was supported by the EU project Hybridge

Stefan Strubbe is with the Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente. Arjan van der Schaft is with the Institute for Mathematics and Computer Science, University of Groningen. s.n.strubbe@math.utwente.nl, a.j.van.der.schaft@math.rug.nl

0-7803-9568-9/05/$20.00 ©2005 IEEE

Definition 2.1: A CPDP is a tuple $(L,V,\nu,W,\omega,F,G,\Sigma,\mathscr{A},\mathscr{P},\mathscr{S})$, where

- L is a set of locations
- V is a set of state variables. With $d(v)$ for $v \in V$ we denote the dimension of variable v. $v \in V$ takes its values in $\mathbb{R}^{d(v)}$.
- W is a set of output variables. With $d(w)$ for $w \in W$ we denote the dimension of variable w. $w \in W$ takes its values in $\mathbb{R}^{d(w)}$.
- $\nu : L \rightarrow 2^V$ maps each location to a subset of V, which is the set of state variables of the corresponding location.
- $\omega : L \rightarrow 2^W$ maps each location to a subset of W, which is the set of output variables of the corresponding location.
- F assigns to each location l and each $v \in \nu(l)$ a mapping from $\mathbb{R}^{d(v)}$ to $\mathbb{R}^{d(v)}$, i.e. $F(l,v) : \mathbb{R}^{d(v)} \rightarrow \mathbb{R}^{d(v)}$. $F(l,v)$ is the vector field that determines the evolution of v for location l (i.e. $\dot{v} = F(l,v)$ for location l).
- G assigns to each location l and each $w \in \omega(l)$ a mapping from $\mathbb{R}^{d(v_1)+\cdots+d(v_m)}$ to $\mathbb{R}^{d(w)}$, where v_1 till v_m are the state variables of location l. $G(l,w)$ determines the output equation of w for location l (i.e. $w = G(l,w)$).
- Σ is the set of communication labels. $\bar{\Sigma}$ denotes the 'passive' mirror of Σ and is defined as $\bar{\Sigma} = \{\bar{a} | a \in \Sigma\}$.
- $\mathscr{A}$ is a finite set of active transitions and consists of five-tuples (l,a,l',G,R), denoting a transition from location $l \in L$ to location $l' \in L$ with communication label $a \in \Sigma$, guard G and reset map R. G is a closed subset of the state space of l. The reset map R assigns to each point in G for each variable $v \in \nu(l')$ a probability measure on the state space (and its Borel sets) of v for location l'.
- $\mathscr{P}$ is a finite set of passive transitions of the form $(l,\bar{a},l',R)$. R is defined on the state space of l (as the R of an active transition is defined on the guard space).
- $\mathscr{S}$ is a finite set of spontaneous transitions and consists of four-tuples (l,λ,l',R), denoting a transition from location $l \in L$ to location $l' \in L$ with jump-rate λ and reset map R. The jump rate λ (i.e. the Poisson rate of the Poisson process of the spontaneous transition) is a mapping from the state space of l to $\mathbb{R}_+$. R is defined on the state space of l as it is done for passive transitions.

Note that the symbol G is used twice; for denoting the output map and for denoting a guard of an active transition. In the rest of this paper, it will directly be clear from the context which use for G is meant.

For a CPDP X with $v \in V_X$, where V_X is the set of state variables of X, we call $\mathbb{R}^{d(v)}$ the state space of state variable v. We call $\{(v=r) | r \in \mathbb{R}^{d(v)}\}$ the valuation space of v and each $(v=r)$ for $r \in \mathbb{R}^{d(v)}$ is called a valuation. We call $\{(v_1 = r_1, v_2 = r_2, \cdots, v_m = r_m) | r_i \in \mathbb{R}^{d(v_i)}\}$, where v_1 till v_m are the variables from $\nu(l)$, the valuation space or (continuous) state space of location l and each $(v_1 = r_1, \cdots, v_m = r_m)$ is called a valuation or (continuous) state of l. A valuation (state) is an unordered tuple (e.g. $(v_1 = 0, v_2 = 1)$ is the same valuation as $(v_2 = 1, v_1 = 0)$). We denote the valuation space of l by $val(l)$. We call $\{(l,x) | l \in L, x \in val(l)\}$ the state space of a CPDP with location set L and valuation spaces $val(l)$. Each state of a CPDP consists of a location (belonging to a discrete set) and a valuation (which takes value in a continuum), therefore we call the state (state space) of a CPDP a hybrid state (hybrid state space). The (continuous) state space of a location l with $\nu(l) = \{v_1, \cdots, v_m\}$ can be regarded as $\mathbb{R}^{d(v_1)+\cdots+d(v_m)}$, because the state space is (topologically) homeomorphic to $\mathbb{R}^{d(v_1)+\cdots+d(v_m)}$ by the homeomorphism $\pi_l : val(l) \rightarrow \mathbb{R}^{d(v_1)+\cdots+d(v_m)}$ defined as $\pi_l((v_1 = r_1, \cdots, v_m = r_m)) = (r_1, \cdots, r_m)$. We use unordered tuples for the valuations (states) because this will turn out to be helpful for the composition operation and for some other definitions and proofs.

The difference between the CPDP model here (i.e. of Definition 2.1) and the CPDP model from [6] is as follows. First, a CPDP location contains both state and output variables, while the CPDP model of [6] does not consider output variables. Second and main difference is that we use guards. This causes non-determinism because a transition may take place anywhere in the guard-area but it is not determined exactly where the transition will take place. In [6] there are no guards but there are boundaries. Using boundaries does not cause non-determinism because then a transition will take place exactly when the boundary is hit. Later we will see that we can model this boundary effect also by using guards together with the socalled maximal progress strategy. The advantages of using guards instead of boundaries are as follows.

1) If a transition jumps into the guard area of another transition, then this other transition is immediately enabled and may therefore immediately be taken. Therefore the CPDP model allows that multiple transitions are taken at the same time instant. In a composition context this means that multiple signals can be broadcast between different components at the same time instant. (The use of this feature is clearly apparent in the Air Traffic Management CPDP-model of [10]). Note that this can not be done in the CPDP model of [6] because there it is not allowed to jump on the boundary of a location.
2) Communication through synchronization of active transitions and through synchronization of active with passive transitions is possible, whereas the CPDP model of [6] only allows synchronization of active with passive transitions.

III. Composition of CPDPs

In the process algebra and concurrent processes literature it is common to define a *parallel composition operator*, normally denoted by $||$. $||$ has as its arguments two processes, say X and Y, of a certain class of processes. The result of the composition operation, denoted by $X||Y$, is again a process that falls within the same class of processes (i.e. the specific class of processes is closed under $||$). The main idea of using this kind of composition operator is that the process $X||Y$ describes the behavior of the composite system that consists of components X and Y (which might interact with each other).

In [7] the composition operator $|^P_A|$ is defined for general transition systems with active and passive transitions. Here, we will use $|^P_A|$ in the context of CPDPs. The sets A and P contain respectively the active and passive events that should synchronize in the composition. Passive events can not happen 'by themselves', but should be triggered by active events from other components. This expressed in rule r2 below. For a full explanation of the use of active and passive events and their interaction, we refer to [7]. The composition rules, which define the operator $|^P_A|$, are given in the Plotkin style, which is common practice in the process algebra literature. This means that we use structural operational rules of the form $\frac{A,B_1}{C}(B_2)$, which should be read as: if A,B_1 and B_2 are true, then this implies that C is true.

$$r1.\frac{l_1 \xrightarrow{a,G_1,R_1} l_1', l_2 \xrightarrow{a,G_2,R_2} l_2'}{l_1|^P_A|l_2 \xrightarrow{a,G_1\times G_2,R_1\times R_2} l_1'|^P_A|l_2'}(a \in A).$$

$$r2.\frac{l_1 \xrightarrow{a,G_1,R_1} l_1', l_2 \xrightarrow{\bar{a},R_2} l_2'}{l_1|^P_A|l_2 \xrightarrow{a,G_1\times val(l_2),R_1\times R_2} l_1'|^P_A|l_2'}(a \notin A).$$

$$r2'.\frac{l_1 \xrightarrow{\bar{a},R_1} l_1', l_2 \xrightarrow{a,G_2,R_2} l_2'}{l_1|^P_A|l_2 \xrightarrow{a,val(l_1)\times G_2,R_1\times R_2} l_1'|^P_A|l_2'}(a \notin A).$$

$$r3.\frac{l_1 \xrightarrow{a,G_1,R_1} l_1', l_2 \not\xrightarrow{\bar{a}}}{l_1|^P_A|l_2 \xrightarrow{a,G_1\times val(l_2),R_1\times Id} l_1'|^P_A|l_2}(a \notin A).$$

$$r4.\frac{l_1 \xrightarrow{\bar{a},R_1} l_1'}{l_1|^P_A|l_2 \xrightarrow{\bar{a},R_1\times Id} l_1'|^P_A|l_2}(\bar{a} \notin P)$$

$$r5.\frac{l_1 \xrightarrow{\bar{a},R_1} l_1', l_2 \xrightarrow{\bar{a},R_2} l_2'}{l_1|^P_A|l_2 \xrightarrow{\bar{a},R_1\times R_2} l_1'|^P_A|l_2'}(\bar{a} \in P).$$

$$r6.\frac{l_1 \xrightarrow{\bar{a},R_1} l_1', l_2 \not\xrightarrow{\bar{a}}}{l_1|^P_A|l_2 \xrightarrow{\bar{a},R_1\times Id} l_1'|^P_A|l_2}(\bar{a} \in P)$$

$$r7.\frac{l_1 \xrightarrow{\lambda_1,R_1} l_1'}{l_1|^P_A|l_2 \xrightarrow{\hat{\lambda}_1,R_1\times Id} l_1'|^P_A|l_2}, \quad r7'.\frac{l_2 \xrightarrow{\lambda_2,R_2} l_2'}{l_1|^P_A|l_2 \xrightarrow{\hat{\lambda}_2,Id\times R_2} l_1|^P_A|l_2'}.$$

In the above rules $G_i \times G_j$ denotes the product space of the guard spaces G_i and G_j, $R_i \times R_j$ denotes the product reset map (consisting of product probability measures) of R_i and R_j and Id is the identity reset map, which leaves each variable that is reset, unaltered with probability one. In rules r7 and r7', $\hat{\lambda}_1$ and $\hat{\lambda}_2$ are defined on the combined state space of locations l_1 and l_2 and equal $\hat{\lambda}_1(x_1,x_2) = \lambda_1(x_1)$ and $\hat{\lambda}_2(x_1,x_2) = \lambda_2(x_2)$, where x_1 and x_2 are states of l_1 and l_2 respectively.

Besides the above rules, we also consider (but do not explicitly state) the rules r3',r4' and r6', which are the mirror rules of r3,r4 and r6 like r2' and r7' are the mirror rules of r2 and r7. We now define the composition of two CPDPs, resulting in a new composite CPDP.

Definition 3.1: If $X = (L_X,V_X,\nu_X,W_X,\omega_X,F_X,G_X,\Sigma,\mathscr{A}_X,\mathscr{P}_X,\mathscr{S}_X)$ and $Y = (L_Y,V_Y,\nu_Y,W_Y,\omega_Y,F_Y,G_Y,\Sigma,\mathscr{A}_Y,\mathscr{P}_Y,\mathscr{S}_Y)$ are two CPDPs that have the same set of events Σ and if we have $V_X \cap V_Y = W_X \cap W_Y = \emptyset$, then $X|^P_A|Y$ is defined as the CPDP $(L,V,\nu,W,\omega,F,G,\Sigma,\mathscr{A},\mathscr{P},\mathscr{S})$, where

- $L = \{l_1|^P_A|l_2 \mid l_1 \in L_X, l_2 \in L_Y\}$,
- $V = V_X \cup V_Y$, $W = W_X \cup W_Y$,
- $\nu(l_1|^P_A|l_2) = \nu(l_1) \cup \nu(l_2)$, $\omega(l_1|^P_A|l_2) = \omega(l_1) \cup \omega(l_2)$,
- $F(l_1|^P_A|l_2,v)$ equals $F_X(l_1,v)$ if $v \in \nu_X(l_1)$ and equals $F_Y(l_2,v)$ if $v \in \nu_Y(l_2)$.
- $G(l_1|^P_A|l_2,w)$ equals $G_X(l_1,w)$ if $w \in \omega_X(l_1)$ and equals $G_Y(l_2,w)$ if $w \in \omega_Y(l_2)$.
- $\mathscr{A}$, $\mathscr{P}$ and $\mathscr{S}$ contain and only contain the transitions that are the result of applying one of the rules r1,r2,r2',r3,r3',r4,r4',r5,r6,r6',r7 and r7', defined above.

The operator $|^P_A|$ is called commutative if for all CPDPs X and Y we have that $X|^P_A|Y$ is isomorphic to $Y|^P_A|X$, where two CPDPs are isomorphic if they can be turned into each other by renaming the locations. The operator $|^P_A|$ is called associative if for all CPDPs X,Y and Z we have that $(X|^P_A|Y)|^P_A|Z$ is isomorphic to $X|^P_A|(Y|^P_A|Z)$.

Theorem 3.2: The composition operator $|^P_A|$ is commutative for all A and P. $|^P_A|$ is associative if and only if for all $a \in \Sigma$ we have: if $\bar{a} \notin P$ then $a \in A$.

Proof: The proof of this theorem in the context of active/passive labelled transition systems can be found on www.cs.utwente.nl/˜strubbesn. The proof can easily be generalized to the context of CPDPs. ■

If we have n CPDPs X_i $(i = 1 \cdots n)$ with events-set Σ that are composed via an associative operator $|^P_A|$, then the order of composition does not influence the resulting CPDP and therefore we can write $X_1|^P_A|X_2|^P_A|\cdots X_{n-1}|^P_A|X_n$ in order to unambiguously (up to isomorphism) denote the resulting composite CPDP.

IV. PDP-SEMANTICS OF CPDPS

Under certain conditions, the state evolution of a CPDP can be modelled as a stochastic process. In this section we give the exact conditions under which this is true. We also prove that the stochastic process may always be chosen of the PDP-type. In order to achieve this result, we first need to make a distinction between guarded CPDP states and unguarded CPDP states.

Definition 4.1: A state (l,x) of a CPDP X is called *guarded*, if there exists an active transition with origin location l such that x is an element of the guard of this transition. A CPDP state is *unguarded* if it is not guarded.

If we execute a CPDP X from some initial hybrid state (l_0,x_0) then the first part of the state trajectory (i.e., the evolution of the state variables in time) and of the output trajectory (i.e. the evolution of the output variables in time) is determined by F_X and G_X respectively. This is the case until the first transition is executed, which might cause a jump (i.e. discontinuity) in the state/output trajectories. We choose that at these points of discontinuity, the state/output trajectories have the *cadlag* property, which means that at these points the trajectories are continuous from the right and have limits from the left. If then at $t = t_1$, X executes a transition which resets the state to an unguarded state x_1, then the value of the state trajectory at $t = t_1$ equals x_1 (and the value of the output trajectory equals the output value of x_1). If the state after reset x_1 is guarded, then it is possible that at the same time t_1 from state x_1 another active transition is executed. If this transition resets the state to an unguarded state x_1', then the value of the state trajectory at t_1 equals x_1'. If this transition resets the state to a guarded state x_1', then another active transition can be executed, etc. We conclude that the CPDP model allows multiple transitions at the same time instant.

Formally, let $E := \{(l,x) | l \in L_X, x \in val(l)\}$ be the state space of CPDP X, where $val(l)$ denotes the space of all valuations for the state variables of location l. The trajectories of X are elements of the space $D_E[0,\infty[$ which is the space of right-continuous E-valued functions on $\mathbb{R}_+$ with left-hand limits. According to [2], a metric can be defined on E such that $(E, \mathscr{B}(E))$, with $\mathscr{B}(E)$ the set of Borel sets of E under this metric, is a Borel space (i.e. a subset of a complete separable metric space) and each Borel set B is such that for each $l \in L_X$, $\{x | (l,x) \in B\}$ (i.e. the restriction of B to l) is a Borel set of the Euclidean state space $val(l)$ of location l. Therefore, the concept of continuity within a location (i.e. for sets $\{(l,x) | x \in val(l)\}$) coincides with the standard (Euclidean) concept of continuity.

The CPDP model exhibits non-determinism. This means that at certain time instants of any execution of a CPDP (starting from some initial state) choices have to be made which are neither deterministic (like a differential equation deterministically determines the state trajectory) nor stochastic (i.e. a probability measure can be used to make a probabilistic choice). These non-deterministic choices are simply unmodelled. We distinguish two sources of non-determinism for the CPDP: 1. The choice when an active transition is taken. 2. The choice which active transition is taken. To resolve non-determinism of type 1, we use, in the line of [4], the *maximal progress* strategy, which means that as soon as the state enters a guard area (i.e. at the first time instant that the state is guarded), an active transition has to be executed. To resolve non-determinism of type 2, we use a socalled *scheduler S* which

1) assigns to each guarded state x a probability measure on the set of all active transitions that have x as an element of their guard (i.e. the set of all active transitions that are allowed to be executed from state x) and
2) assigns to each pair $(x, \bar{a})$, with x any state and $\bar{a} \in \bar{\Sigma}$ such that there is a $\bar{a}$-transition at the location of x, a probability measure on the set of all $\bar{a}$-transitions at the location of x.

In other words, if an active transition has to be executed from state x, S probabilistically chooses which active transition is executed and if an active a triggers a $\bar{a}$-transition, then S probabilistically chooses which $\bar{a}$-transition is executed.

For identifying the stochastic process of a CPDP, we only look at *closed* CPDPs, which are CPDPs that have no passive transitions. Closed CPDPs are called closed because we assume that they represent the whole system (i.e. no more other component-CPDPs will be added). Therefore closed CPDPs should have no passive transitions because passive transitions can only be executed when another component triggers it (via an active transition). The order of finding the stochastic behavior of the composite system is therefore: first compose the different components. Then remove all passive transitions of the resulting CPDP. This results in a closed CPDP where, under maximal progress and scheduler S, all choices for the execution of the CPDP are made probabilistically. One could question whether the evolution of the state can, for closed CPDPs, be modelled as a stochastic process. We can state a condition on the CPDP under which this is certainly not possible: if with non-zero probability we can reach a guarded state x where with non-zero probability an infinite sequence of active transitions can be chosen such that each transition resets the state within the guard of the next transition, then the trajectory of this execution deadlocks (i.e. time does not progress anymore after reaching x at some time $\hat{t}$ and therefore the trajectory is not defined for time instants after time $\hat{t}$). Trajectories of stochastic processes do not deadlock like this, therefore this state evolution cannot be modelled by a stochastic process.

In order to find the stochastic process of a closed CPDP, we would first like to state conditions on a CPDP, which guarantee that the probability that an execution deadlocks (i.e. comes at a point where time does not progress anymore) is zero.

A. The stochastic process of a closed CPDP

Suppose we have a closed CPDP X with location set L_X and active transition set $\mathscr{A}_X$. The CPDP operates under maximal progress and under scheduler S. We write $S_x(\alpha)$ for the probability that active transition α is taken when an active transition is executed at state x. We assume that the CPDP has no spontaneous transitions. The case 'with spontaneous transitions' is treated at the end of this section.

We call the jump of a CPDP from the current state to another unguarded state via a sequence of active transitions a hybrid jump. We call the number of active transitions involved in a hybrid jump the multiplicity of the hybrid jump. For example, if at state x_1 a transition α is taken to x_1', which lies in the guard of transition β, and immediately transition β is taken to an unguarded state x_1'', then this hybrid jump from x_1 to x_1'' has multiplicity two.

We need to introduce the concept of total reset map. $R_{tot}(B,x)$ denotes the probability of jumping into $B \in \mathscr{B}(E)$ when an active transition takes place at state x. (Here, B may contain both guarded and unguarded states). We have that

$$R_{tot}(B,x) = \sum_{\alpha \in \mathscr{A}_{l_x \to}} [S_x(\alpha) R_\alpha(B \cap val(l'_\alpha), x)],$$

where $\mathscr{A}_{l_x \to}$ is the set of all active transitions that leave the location of x. We define the total guard $G_{tot,l}$ of location l as the union of the guards of all active transitions with origin location l. It can be seen now that for the stochastic executions (i.e. generating trajectories during simulation) of X it is enough to know R_{tot} and $G_{tot,l}$ (for all $l \in L_X$) instead of $\mathscr{A}_X$: a trajectory that starts in (l_0, x_0) evolves until it hits G_{tot,l_0} at some state (l_0, x_1). From x_1 we determine the target state (l_1, x_1') of the (first step of the) hybrid jump by drawing a sample from $R_{tot}(\cdot, x_1)$. If x_1' is unguarded, the next piecewise deterministic part of the trajectory is determined by the differential equations of the state variables of location l_1 until G_{tot,l_1} is hit. If x_1' is guarded, we directly draw a new target state (l_1', x_1'') from $R_{tot}(\cdot, x_1')$, etc. Therefore, if two closed CPDPs that are isomorphic except for the active transition set, and they have the same total reset map and the same total guards, then the stochastic behaviors (concerning the state trajectories) of the two CPDPs are the same and consequently if some stochastic process models the state evolution of one CPDP, then it also models the state evolution of the other CPDP.

We will now present the algorithm that can convert CPDPs to CPDPs of the type of [6]. The algorithm consists of three parts. First, we show how a transition can be split into a stable and unstable part such that the stochastic behavior does not change. Second, we show, by using the distinction between stable and unstable transitions, how chains of transitions can be converted into a single transition without changing the stochastic behavior. Third, we show how the results from the first two steps can be used to determine the PDP that models the behavior of the original CPDP.

a) Finding the stable and unstable parts of an active transition: Take any $\alpha \in \mathscr{A}_X$. We now show how to split up α in a stable part α_s and an unstable part α_u such that if we replace α by α_s and α_u, then the stochastic behavior of X does not change.

We define G_{α_s} as the set of all $x \in G_\alpha$ (i.e. all x in the guard of α) such that $R_\alpha(val_s(l'_\alpha), x) \neq 0$, where $val_s(l'_\alpha)$ is the unguarded part of the state space of the target location of α. Then for all $x \in G_{\alpha_s}$ we define

$$R_{\alpha_s}(B,x) := \frac{R_\alpha(B \cap val_s(l'_\alpha), x)}{R_\alpha(val_s(l'_\alpha), x)},$$

$$S_x(\alpha_s) := S_x(\alpha) R_\alpha(val_s(l'_\alpha), x).$$

The scheduler works on α_s as $S_x(\alpha_s)$ (as defined above).

We define G_{α_u} as the set of all $x \in G_\alpha$ such that $R_\alpha(val_u(l'_\alpha), x) \neq 0$. For all $x \in G_{\alpha_u}$ we define

$$R_{\alpha_u}(B,x) := \frac{R_\alpha(B \cap val_u(l'_\alpha), x)}{R_\alpha(val_u(l'_\alpha), x)},$$

$$S_x(\alpha_s) := S_x(\alpha) R_\alpha(val_u(l'_\alpha), x).$$

The scheduler works on α_u as $S_x(\alpha_u)$ (as defined above).

It can be seen that replacing α by α_s and α_u does not change the total reset map.

b) Resolving hybrid jumps of multiplicity greater than one: For any $n \in \mathbb{N}$ we will now define T_s^n and T_u^n. T_s^n is a set of stable transitions representing hybrid jumps of multiplicity n and T_u^n is a set of unstable transitions representing hybrid jumps of multiplicity n. A stable transition is a transition that always jumps to the unguarded state space of the target location. An unstable transition always jumps to the guarded state space. A stable transition is stable in the sense that after the hybrid jump caused by the transition, no other hybrid jump will happen immediately and therefore we are sure that a stable transition will not cause an explosion of active transitions (i.e. a hybrid jump of multiplicity infinity). An unstable transition does not necessarily need to induce such a blow up of active transitions, but potentially it can.

We define T_s^1 as the set of all active transitions α_s (with $\alpha \in \mathscr{A}_X$) such that $G_{\alpha_s} \neq \emptyset$ and we define T_u^1 as the set of all active transitions α_u (with $\alpha \in \mathscr{A}_X$) such that $G_{\alpha_u} \neq \emptyset$.

We introduce the following notations. $P_x(B \circ \beta \circ \alpha)$ denotes the probability that, given that an active jump takes place at state x, transition α is executed followed directly by transition β jumping into the set $B \in \mathscr{B}(val(l'_\beta))$. It can be seen that

$$P_x(B \circ \beta \circ \alpha) = S_x(\alpha) \int_{\hat{x} \in G_\beta} S_{\hat{x}}(\beta) R_\beta(B, \hat{x}) \mathrm{d}R_\alpha(\hat{x}, x).$$

We will show how the sets T_s^n and T_u^n can inductively be determined. Suppose the sets T_s^{n-1} and T_u^{n-1} and T_s^1 and T_u^1 are given. Now, for any $\alpha \in T_u^{n-1}$, $\beta \in T_s^1 \cup T_u^1$ such that $l'_\alpha = l_\beta$, we define $G_{\beta\circ\alpha}$ as all $x \in G_\alpha$ such that $R_\alpha(G_\beta, x) \neq 0$. Then, for all $x \in G_{\beta\circ\alpha}$ we define

$$S_x(\beta \circ \alpha) := P_x(val(l'_\beta) \circ \beta \circ \alpha),$$

$$R_{\beta\circ\alpha}(B,x) := \frac{P_x(B \circ \beta \circ \alpha)}{S_x(\beta \circ \alpha)}.$$

If $G_{\beta\circ\alpha} \neq \emptyset$ and $\beta \in T_s^1$ then we add transition $\beta \circ \alpha$, with guard, reset map and scheduler as above, to T_s^n. If $G_{\beta\circ\alpha} \neq \emptyset$ and $\beta \in T_u^1$ then we add transition $\beta \circ \alpha$, with guard, reset map and scheduler as above, to T_u^n.

c) Finding the PDP that models the state evolution of the CPDP: If we define, for $z \in \{s, u\}$ and $B \in \mathscr{B}(E)$,

$$R^n_{tot,z}(B,x) := \sum_{\{\alpha \in T_z^n | l_\alpha = l_x\}} [S_x(\alpha) R_\alpha(B \cap val(l'_\alpha), x)],$$

with $B \cap val(l'_\alpha)$ sloppy notation for $\{x | x \in val(l'_\alpha), (l'_\alpha, x) \in B\}$, then it can be seen that for any $n \in \mathbb{N}$ we have

$$R_{tot}(B,x) = \sum_{i=1}^{n} [R^i_{tot,s}(B,x)] + R^n_u(B,x),$$

with other words, if X^n is isomorphic to CPDP X, except that the active transition set of X^n equals $T_s^1 \cup T_s^2 \cup \cdots \cup T_s^n \cup T_u^n$ (which need not be isomorphic to $\mathscr{A}_X$), then the total reset maps of X and X^n are the same for all n.

We are now ready to state the theorem which gives necessary and sufficient conditions on the CPDP such that the state evolution can be modelled by a stochastic process. Also, the theorem says that if the state evolution can be modelled by a stochastic process, then it can be modelled by a stochastic process from the class of PDPs. The proof of the theorem makes use of the results from [8].

Theorem 4.2: Let X^n be derived from X as above. Let $R^n_{tot,s}$ denote the total stable reset map of X^n. The state evolution of X can be modelled by a stochastic process if and only if $R(E,x) := \lim_{n\to\infty} R^n_{tot,s}(E,x) = 1$ for all $x \in E_u$, with E_u the guarded part of E. If this condition is satisfied, then the PDP with the same state space as X, with invariants $E^0_l = val(l)\backslash G_{tot,l}$ and with transition measure $Q(B,x) = R(B,x)$, models the state evolution of X.

Proof: From the text above and from the results of [8], it is clear that if $R(E,x) = 1$ for all x, then the PDP suggested by the theorem models the state evolution of X. If for some $x \in E$, $R(E,x) < 1$, then it can be seen that this must mean that there exists a hybrid jump with multiplicity infinity such that the probability of this hybrid jump at x is greater than zero. This means that (from x) there is a deadlock probability (i.e. time does not progress anymore) greater than zero, which means that the state evolution of X cannot be modelled by a stochastic process (as we saw before). ∎

Corollary 4.3: If for some $n \in N$ we have that $T^n_u = \emptyset$, then the multiplicity of the hybrid jumps of X is bounded by n and the state of X exhibits a PDP behavior, with the same PDP as the corresponding PDP of X^n (which can be constructed according to [8] because all hybrid jumps of X^n have multiplicity one).

B. The case including spontaneous transitions

Now we treat the case where there are also spontaneous transitions present. Let X be a CPDP without passive and spontaneous transitions and let $\hat{X}$ be an isomorphic copy of X together with a set of spontaneous transitions $\mathscr{S}_{\hat{X}}$. Suppose that the multiplicity of the hybrid jumps of X is bounded by n. Let $\hat{X}^n$ be an isomorphic copy of X^n together with the following spontaneous transitions: for any spontaneous transition $(l,\lambda,l',R) \in \mathscr{S}_{\hat{X}}$ we add to $\hat{\mathscr{S}}$, which denotes the set of spontaneous transitions of $\hat{X}^n$, the transition $(l,\lambda,L,\hat{R})$, where, for $B \in \mathscr{B}(E)$,

$$\hat{R}(B,x) := R(B \cap Inv_s(l'),x) + \sum_{\{\alpha \in \mathscr{A}_{X^n} | l_\alpha = l\}} \int_{\hat{x} \in G_\alpha} S_{\hat{x}}(\alpha) R_\alpha(B \cap val(l'_\alpha)) \mathrm{d}R(\hat{x},x).$$

Note that all transitions from $\mathscr{A}_{X^n}$ are stable. Also note that $(l,\lambda,L,\hat{R})$ is not a standard CPDP transition, but a transition that represents a Poisson process in location l with jump-rate λ and with reset map $\hat{R}$, which can jump to multiple locations. Therefore we write L instead of l' in the tuple of the transition.

It is known that the superposition of two (or more) Poisson processes is again a Poisson process (see, in the context of CPDP, [8] for a proof of this result). This means that if we combine all spontaneous transitions of $\hat{X}^n$ with origin location l to one spontaneous transition $(l,\lambda_l,L,\hat{R}_{tot,l})$, with

$$\lambda_l(x) = \sum_{\alpha \in \hat{\mathscr{S}}_{l\rightarrow}} \lambda_\alpha(x),$$

and

$$\hat{R}_{tot,l}(B,x) = \sum_{\alpha \in \hat{\mathscr{S}}_{l\rightarrow}} \left(\frac{\lambda_\alpha(x)}{\lambda_l(x)} R_\alpha(B,x) \right),$$

and if we replace all spontaneous transitions by these combined spontaneous transitions, then the stochastic behavior (concerning the evolution of the state) will not change. Now it can be easily seen that if we add jump rate $\lambda(l,x) = \lambda_l(x)$ to the PDP that models the state evolution of X and we let, for unguarded states (l,x), the transition measure $Q(B,(l,x)) = \hat{R}_{tot,l}(B,x)$, then this PDP will model the state evolution of $\hat{X}$.

For a CPDP with spontaneous transitions, the condition '$R(E,x) := \lim_{n\to\infty} R^n_{tot,s}(E,x) = 1$ for all $x \in E_u$' should be replaced by '$R(E,x) := \lim_{n\to\infty} R^n_{tot,s}(E,x) = 1$ for all $x \in E$' because a spontaneous transition might jump to a state $x \in E_g$ in the guarded part of the state space, and then directly from this state a chain of active transitions is executed. To assure that this chain behaves 'PDP-like' we need $R(E,x) = 1$.

We do not have enough space here to illustrate the conversion algorithm. We refer to [10] for illustrations. There, this algorithm is applied to a composite machine repair shop system and to a composite Air Traffic Management system.

V. Conclusions

In this paper we have presented an extension to the CPDP framework of [6]. This extension gives richer interaction possibilities in two ways. First, communication through shared active events is possible and second, communication via multiple signals at the same time instant is possible.

Because of using guards, the CPDP model diverges from the PDP model. We have shown in this paper that by using a scheduler and the maximal progress strategy to resolve non-determinism, the behavior of a CPDP can, under certain conditions, still be modelled through a PDP. We have given an algorithm which, if it terminates, gives this corresponding PDP.

References

[1] M. H. A. Davis, "Piecewise Deterministic Markov Processes: a general class of non-diffusion stochastic models," *Journal Royal Statistical Soc. (B)*, vol. 46, pp. 353–388, 1984.

[2] ——, *Markov Models and Optimization*. London: Chapman & Hall, 1993.

[3] M. H. C. Everdij and H. A. P. Blom, "Petri-nets and hybrid-state markov processes in a power-hierarchy of dependability models," in *Preprints Conference on Analysis and Design of Hybrid Systems ADHS 03*, 2003, pp. 355–360.

[4] H. Hermanns, *Interactive Markov Chains*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2428.

[5] M. H.-H. L. Logrippo, M. Faci, "An introduction to lotos: Learning by examples," *Comp. Networks and ISDN Systems*, vol. 23(5), pp. 325–342, 1992.

[6] S. N. Strubbe, A. A. Julius, and A. J. van der Schaft, "Communicating Piecewise Deterministic Markov Processes," in *Preprints Conference on Analysis and Design of Hybrid Systems ADHS 03*, 2003, pp. 349–354.

[7] S. N. Strubbe and R. Langerak, "A composition operator for complex control systems," 2005, submitted to Formal Methods conference 2005.

[8] S. N. Strubbe and A. J. van der Schaft, "Stochastic equivalence of CPDP-automata and Piecewise Deterministic Markov Processes," 2005, accepted for the IFAC world congress 2005.

[9] S. N. Strubbe and A. J. van der Schaft, "Bisimulation for Communicating Piecewise Deterministic Markov Processes (CPDPs)," in *HSCC 2005*, ser. Lecture Notes in Computer Science, vol. 3414. Springer, 2005, pp. 623–639.

[10] S. N. Strubbe, "Compositional modelling of stochastic hybrid systems," Twente University, Phd. Thesis, 2005.