COMMUNICATING PIECEWISE DETERMINISTIC MARKOV PROCESSES

S.N. Strubbe [1] A.A. Julius A.J. van der Schaft [1]

Abstract: In this paper we introduce CPDPs (Communicating Piecewise Deterministic Markov Processes) as an automata formalism for compositional specification of hybrid systems of the type PDP. A CPDP can be seen as an automaton representation of a PDP, with the extra possibility of interaction with other processes via a new concept that we call passive transitions. In order to compose two CPDPs, we define a composition operator on CPDPs and we show that the class of CPDPs is closed under this operator.

Keywords: Piecewise Deterministic Markov Processes, hybrid systems, compositional specification.

1. INTRODUCTION

In this paper we introduce CPDPs. This is an automata formalism which can be used for compositional specification of stochastic hybrid systems of the type PDP.

In 1984 Davis (Davis 1984) introduced the class of PDPs (Piecewise Deterministic Markov Processes). A PDP is a stochastic process of hybrid type. This means that the stochastic process concerns both a discrete location and a continuous variable. The class of PDPs was recognized as a very wide class holding many types of stochastic hybrid systems. PDPs are considered very useful in (Everdij and Blom 2000) within the Air Traffic Management context.

Hybrid systems have been studied intensively from the nineties on by both mathematicians and computer scientists. The concept of compositional specification is seen as an important notion for many years in computer science, but in mathematical systems theory this notion did not have much attention until the field of hybrid systems became popular. Many real life hybrid systems have a very complex nature where many different parts are interacting with each other. Therefore, there is a need for compositional specification in hybrid systems (van der Schaft and Schumacher 2001), such that a system can be specified by specifying its constituent parts.

In the literature, only few efforts have been made to find a graphical formalism for representing PDP systems. In computer science, automata and Petri nets are seen as two of the most important graphical formalisms. In (Everdij and Blom 2000) it was proven that the class of DCPNs (Dynamically Coloured Petri Nets) is equivalent to the class of PDPs. This means that DCPN gives a graphical tool to model PDP systems with the compositional power of Petri nets. In this paper we take an automata approach, rather than a Petri net approach, for compositional specification of PDPs.

For that purpose we introduce the automata framework CPDP (Communicating Piecewise Deterministic Markov Processes) for modelling PDP systems. This framework is connected to the discrete event systems IMC (Interactive Markov Chains) framework of Hermanns (Hermanns 1999). In IMC there are two kind of transitions, interactive transitions and Markov transitions. Interactive transitions can be used for interaction or synchronization between systems, but Markov transitions are autonomous and can not be used for interaction. In IMC we find a formalism for

[1] Funded by EU project Hybridge

compositional specification of a class of (non-hybrid) stochastic systems. Compositional specification of (non-stochastic) hybrid systems is done for example in (Alur *et al.* 1995) and (Lynch *et al.* 2001).

In CPDP there are two kinds of autonomous transitions, Markov and Boundary-hit transition, and there is one kind of interactive transitions, called Passive transitions. The notion of interaction in CPDP, which will be treated in section 3, is what we call one-way synchronization, see also (Julius *et al.* 2003). This notion is different from the ones in (Alur *et al.* 1995) and (Hermanns 1999) where there is two-way synchronization. Communication in (Lynch *et al.* 2001) is also of a one-way kind. Typical for one-way synchronization is that the active (or output) partner can influence the passive (or input) partner, but the passive partner can not influence the behavior of the active partner. In many communication or interaction systems this kind of one-way synchronization seems to be very natural.

2. DEFINITIONS AND NOTATIONS

In this section we introduce the CPDP formalism. First we will formally define its structure and after that we will explain how the execution of a CPDP takes place.

Definition 1. A CPDP $\mathcal{A}$ is a 9-tuple $(L, d, Inv, A, C, B, M, P, G)$, where

- L is a countable set of locations.
- $d : L \to \mathbb{N}$ is a mapping, which maps each location to the dimension of the continuous state space in that location.
- Inv maps each location to its invariant set. This means that for each $l \in L$, $Inv(l)$ is an open subset of $\mathbb{R}^{d(l)}$. We also define the boundary set $\partial Inv(l) := \overline{Inv(l)} \backslash Inv(l)$ of $Inv(l)$.
- A is the set of labels.
- C is the transition-choice function. $C(b, l, x) \in [0, 1]$ is the probability of executing the boundary-hit transition b (see next item) from the boundary state (l, x). $C(b, l, x)$ is defined on all $l \in L$ and all $x \in \partial Inv(l)$ and all $b \in B$ that are outgoing transitions of l. Furthermore $\sum_{b \in B_{l\to}} C(b, l, x) = 1$, where $B_{l\to}$ is the set of all elements of B that are outgoing transitions of l.
- B is the set of boundary-hit transitions. Each element b of B is a quadruple (l, a, l', R), where l is the origin location, a is the label of the jump, l' is the target location, and R is the reset map of the jump. $R(A, x) \in [0, 1]$ is the probability of jumping into the set A when the transition b is taken from boundary state x. $R(A, x)$ is defined for all $x \in \partial Inv(l)$ with $C(b, l, x) > 0$ and for all Borel subsets A of $Inv(l')$.
- M is the set of Markov transitions. Each element m of M is a pentuple (l, a, l', R, λ), where l is the origin location, a is the label of the jump, l' is the target location, R is the reset map of the jump, and λ is the jump rate. $R(A, x) \in [0, 1]$ is the probability of jumping into the set A when the transition m is taken from state x. $R(A, x)$ is defined for all $x \in Inv(l)$ and for all Borel subsets A of $Inv(l')$. $\lambda : Inv(l) \to \mathbb{R}_+$ is a bounded Borel measurable function, and determines the rate of jumping of the process.
- P is the set of passive transitions. Each element p of P is a quadruple (l, a, l', R), where l is the origin location, a is the label of the jump, l' is the target location, and R is the reset map of the jump. $R(A, x) \in [0, 1]$ is the probability of jumping into the set A when the transition b is taken from state x. $R(A, x)$ is defined for all $x \in Inv(l)$ and for all Borel subsets A of $Inv(l')$.
- G determines the flow of the continuous states within the locations. For each $l \in L$, $G(l)$ is a locally Lipschitz continuous function from $\mathbb{R}^{d(l)}$ to $\mathbb{R}^{d(l)}$, and is the vectorfield for the continuous flow in location l.

A CPDP $\mathcal{A}$ as defined above, generates a stochastic process. To give insight to the execution of a CPDP process, we will now describe how sample paths of this stochastic process can be generated. As we will see later, the passive transitions play a role in communication with the outside world. For the generation of a sample path, we assume that the outside world is silent, in other words, no communication takes place and therefore, the passive transitions play no role in this sample path generation.

Execution of a CPDP

We assume that an initial hybrid state $\xi_0 = (l_0, x_0)$ is given. The flow of the continuous state x in location l_0 is determined by the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t} x = G_{l_0}(x),$$

$G(l)$ is written here as G_l. Suppose that $x(t)$ reaches the boundary $\partial Inv(l_0)$ at time τ_b and suppose that location l_0 has n outgoing Markov transitions. During the continuous flow, for every outgoing Markov transition, a stochastic process of the Poisson type is active, which can cause a jump to another hybrid state. The probability density functions of these processes equal

$$\lambda_i(x(t))\mathrm{e}^{-\int_0^t \lambda_i(x(t))\mathrm{d}t},$$

where λ_i is the jump rate of the i-th outgoing Markov transition. For each outgoing Markov transition m_i, we draw a sample τ_i from its Poisson probability distribution. This means that m_i would cause a jump at time τ_i if no other jump occurred before τ_i, therefore the most relevant Markov transition is the one corresponding to

$$\tau_M := \min_{i=1..n} \tau_i.$$

Now there are two possibilities. Firstly, if $\tau_b < \tau_M$, the boundary is reached before any Markov transition is about to be executed, which means that a boundary-hit transition is executed at time τ_b. Secondly, if $\tau_M < \tau_b$, the Markov transition corresponding to τ_M causes a jump at time τ_M before the boundary is reached.

Boundary-hit transition A boundary-hit transition at time τ_b from the boundary state $x(\tau_b) \in \partial Inv(l_0)$ is executed as follows. It could be that multiple boundary-hit transitions are active in state $x(\tau_b)$, therefore we use the choice function C to choose one of the active transitions. We draw a sample $b \in B_{l_0\rightarrow}$ from the probability measure determined by $C(\cdot, l_0, x(\tau_b))$. Now, $b = (l_0, a_b, l'_b, R_b)$ is the transition that will be executed. The target location is l'_b and the target state x' within l'_b is drawn from the probability measure $R_b(\cdot, x(\tau_b))$. With the new hybrid state (l'_b, x') at time τ_b, we can repeat the algorithm above to continue the sample path.

Markov transition A Markov transition $m = (l_0, a_m, l'_m, R_m, \lambda_m)$ at time τ_M from the invariant state $x(\tau_M) \in Inv(l_0)$ is executed as follows. The target location is l'_m and the target state x' within l'_m is drawn from the probability measure $R_m(\cdot, x(\tau_M))$. With the new hybrid state (l'_m, x') at time τ_M, we can repeat the algorithm above to continue the sample path.

With the sample path description above, we informally described the stochastic processes that correspond to the CPDPs. From the definition of PDPs in (Davis 1984) it can be seen that the stochastic processes of CPDPs are much like the PDP stochastic processes. In fact it was shown that the class of CPDPs and the class of PDPs generate the same class of stochastic processes (Strubbe 2003).

In the next section we use the following shorthand notation. We write $l \xrightarrow{a,R} l'$, $l \overset{a,R}{\dashrightarrow} l'$, $l \xmapsto{a,R,\lambda} l'$ to denote the existence of respectively boundary-hit, passive and Markov jumps from l to l' with label a, reset map R and, in the case of a Markov jump, jump rate λ. Furthermore, $l \rightarrow l'$ denotes the existence of a boundary-hit jump from l to l' and $l \rightarrow$ denotes the existence of a boundary-hit jump outgoing from l. For Markov and passive jumps we use equivalent notations. Finally, we also have negated versions like $l \not\rightarrow$, which means that l has no outgoing boundary-hit transitions.

3. COMPOSITION OF CPDPS

In this section we introduce a composition operator $||$ on CPDPs. In CPDPs, communication or interaction takes place via the passive transitions. In a context with two composed CPDPs, one of the CPDPs can execute a passive transition with label a if and only if at the same time the other CPDP executes an active event with label a. We call all boundary-hit and all Markov transitions active transitions. The execution of an active transition always happens independently from the other systems in the composition context. This notion of composition is formally defined below.

Definition 2. Suppose the CPDPs $\mathcal{A}_i = (L_i, d_i, Inv_i, A, C_i, B_i, M_i, P_i, G_i)$ are given for $i = 1, 2$. Then the composition $\mathcal{A} = \mathcal{A}_1 || \mathcal{A}_2$ is characterized by the 9-tuple $(L, d, Inv, A, C, B, M, P, G)$, where

$$\begin{aligned} L &= L_1 \times L_2, \\ d(l_1, l_2) &= d_1(l_1) + d_2(l_2), \\ Inv(l_1, l_2) &= \{(x_1, x_2) | x_1 \in Inv(l_1) \\ &\qquad \text{and } x_2 \in Inv(l_2)\}, \end{aligned}$$

and

$$G(l_1, l_2) = \begin{pmatrix} G_1(l_1) \\ G_2(l_2) \end{pmatrix},$$

which means that the vectorfield $G(l_1, l_2)$ of location (l_1, l_2) is the stacked vector containing the vectorfields of the original locations l_1 and l_2. In the rules **r1** till **r6** below, l_1 and l'_1 are in L_1 and l_2 and l'_2 are in L_2.

$b \in B$, $m \in M$ and $p \in P$ if b, m and p can be derived from the following rules,

$$\textbf{r1.}\ \frac{l_1 \xrightarrow{a,R_1} l'_1,\ l_2 \overset{a}{\not\dashrightarrow}}{(l_1, l_2) \xrightarrow{a,R} (l'_1, l_2)},\quad \textbf{r2.}\ \frac{l_1 \xrightarrow{a,R_1} l'_1,\ l_2 \overset{a,R_2}{\dashrightarrow} l'_2}{(l_1, l_2) \xrightarrow{a,R} (l'_1, l'_2)},$$

$$\textbf{r3.}\ \frac{l_1 \xmapsto{a,R_1,\lambda_1} l'_1,\ l_2 \overset{a}{\not\dashrightarrow}}{(l_1, l_2) \xmapsto{a,R,\lambda} (l'_1, l_2)},\quad \textbf{r4.}\ \frac{l_1 \xmapsto{a,R_1,\lambda_1} l'_1,\ l_2 \overset{a,R_2}{\dashrightarrow} l'_2}{(l_1, l_2) \xmapsto{a,R,\lambda} (l'_1, l'_2)},$$

$$\textbf{r5.}\ \frac{l_1 \overset{a,R_1}{\dashrightarrow} l'_1,\ l_2 \overset{a}{\not\dashrightarrow}}{(l_1, l_2) \overset{a,R}{\dashrightarrow} (l'_1, l_2)},\quad \textbf{r6.}\ \frac{l_1 \overset{a,R_1}{\dashrightarrow} l'_1,\ l_2 \overset{a,R_2}{\dashrightarrow} l'_2}{(l_1, l_2) \overset{a,R}{\dashrightarrow} (l'_1, l'_2)}.$$

We will now specify R in **r1** till **r6**. Here, we make use of the analysis result that for two measures M_1 on Σ_1 and M_2 on Σ_2, there is a unique measure M on $\Sigma_1 \times \Sigma_2$ (called the product measure), such that $M(A_1 \times A_2) = M_1(A_1)M_2(A_2)$ for all Borel subsets A_1 and A_2. We also make use of the indicator function $1_A(x)$ which equals one if $x \in A$ and zero if $x \notin A$. Then in **r1** till **r6**, R is the product measure where $R(A_1 \times A_2, (x_1, x_2)) = R_1(A_1, x_1) 1_{A_2}(x_2)$ (case **r1**, **r3**, **r5**) and $R(A_1 \times A_2, (x_1, x_2)) = R_1(A_1, x_1) R_2(A_2, x_2)$ (case **r2**, **r4**, **r6**) for all $x_1 \in \partial Inv(l_1)$ (case **r1**,**r2**) or all $x_1 \in Inv(l_1)$ (case **r3**, **r4**, **r5**, **r6**), all $x_2 \in Inv(l_2)$ and all Borel subsets A_1 of $Inv(l_1')$ and A_2 of $Inv(l_2)$. Furthermore, in both **r3** and **r4**, $\lambda(x_1, x_2) = \lambda_1(x_1)$ for all $x_1 \in Inv(l_1)$ and all $x_2 \in Inv(l_2)$.

Beside the rules **r1** till **r6**, there are the rules **r1'** till **r6'** which are the mirrored versions of **r1** till **r6**. This means that

$$\textbf{r1'}.\ \frac{l_2 \xrightarrow{a,R_2} l_2', l_1 \overset{a}{\not\dashrightarrow}}{(l_1,l_2) \xrightarrow{a,R} (l_1,l_2')}, \textbf{r2'}.\ \frac{l_2 \xrightarrow{a,R_2} l_2', l_1 \overset{a,R_1}{\dashrightarrow} l_1'}{(l_1,l_2) \xrightarrow{a,R} (l_1',l_2')},$$

etc. Transitions are elements of B, M or P, if and only if they follow from the rules **r1** till **r6** and **r1'** till **r6'**.

At last, we have to specify C. Take any $l = (l_1, l_2) \in L$ and any $b : (l_1, l_2) \to (l_1', l_2') \in B$. b has been derived from one of the following four cases:

$$\textbf{c1:}\ \underbrace{l_1 \to l_1'}_{t_1}, l_2 \not\dashrightarrow, l_2 = l_2'\ (\textbf{r1}),$$

$$\textbf{c2:}\ \underbrace{l_1 \to l_1'}_{t_2}, l_2 \dashrightarrow l_2'\ (\textbf{r2}),$$

$$\textbf{c3:}\ \underbrace{l_2 \to l_2'}_{t_3}, l_1 \not\dashrightarrow, l_1 = l_1'\ (\textbf{r1'}),$$

$$\textbf{c4:}\ \underbrace{l_2 \to l_2'}_{t_4}, l_1 \dashrightarrow l_1'\ (\textbf{r2'}).$$

Now for all $(x_1, x_2) \in \partial Inv(l_1, l_2)$, C is defined as

$$C(b, l, (x_1, x_2)) =$$

$$\begin{cases} C_1(t_1, l_1, x_1) & ;\ \textbf{c1}, x_1 \in \partial Inv_1(l_1), \\ & \quad x_2 \in Inv_2(l_2), \\ C_1(t_2, l_1, x_1) & ;\ \textbf{c2}, x_1 \in \partial Inv_1(l_1), \\ & \quad x_2 \in Inv_2(l_2), \\ C_2(t_3, l_2, x_2) & ;\ \textbf{c3}, x_1 \in Inv_1(l_1), \\ & \quad x_2 \in \partial Inv_2(l_2), \\ C_2(t_4, l_2, x_2) & ;\ \textbf{c4}, x_1 \in Inv_1(l_1), \\ & \quad x_2 \in \partial Inv_2(l_2), \\ \text{undefined} & ;\ x_1 \in \partial Inv_1(l_1), \\ & \quad x_2 \in \partial Inv_2(l_2), \\ 0 & ;\ \text{else.} \end{cases}$$

We will say some words about the meaning of the rules **r1,r1'** till **r6,r6'**. If one agent can execute an active event and the other agent does not have a matching passive partner, then in the composition context, the active agent executes the transition while the other agent stays in the same location (rules **r1,r1',r3,r3'**). If the first agent can execute an active transition and the second agent has a matching passive partner, then in the composition context, both agents execute respectively the active and passive transition at the same time (rules **r2,r2',r4,r4'**). If the first agent has a passive transition with label a and the second agent has no passive transition with label a, then the composed system has a passive transition with label a outgoing from the joint location, which gives the possibility to interact with other systems in another composition context (rules **r5,r5'**). If both agents have a passive transition with the same label, then the composed system also has a passive transition with this label. This means that both agents can execute the passive transitions at the same time in another composition context where a third party executes an active transition with the same label (rules **r6,r6'**).

We see in Definition 2 that C is undefined for all (x_1, x_2) where both x_1 and x_2 are boundary points. We call such a state a *double boundary state*. In the execution of the system, these points play a role only when the two components reach their boundaries at exactly the same time τ_b. It is not evident what to do in such a situation. Two boundary-hit transitions should be executed at the same time. These transitions may have different labels and then a simultaneous execution of these transitions gives problems from a compositionality point of view: If a third party has both labels available in passive transitions, which passive transition should be chosen? However, for many composed systems these problems will not be present because the probability that two separate agents reach their boundary at exactly the same time will be zero. For the rest of this paper, we leave the choice of what to do in double boundary states open and we say that the composed CPDP is undefined on the double boundary states. We now state a theorem which says that the composition of two CPDP behaves as a CPDP, except on the double boundary states.

Theorem 1. If $\mathcal{A}_1$ and $\mathcal{A}_2$ are CPDPs, then $\mathcal{A} = \mathcal{A}_1 || \mathcal{A}_2 = (L, d, Inv, A, C, B, M, P, G)$ is a CPDP which is undefined on double boundary states.

Proof: That L, d, Inv, A and G are correct and compatible CPDP elements, follows directly from the definitions of CPDP and the $||$ operator. Concerning P, we see that the only non-trivial elements from passive transitions are the reset maps R. The reset maps of elements of P are defined in rules **r5** and **r6** and their mirrors. From these rules we see that every reset map is defined on all elements of the invariant of the target location and every reset map correctly defines a probability measure on the Borel sets of the invariant set of the target location, this means that all CPDP conditions on P are met. For M, the same story is true for the reset maps and the jump rates λ are evidently correct. For B there is a different story. As far as the reset maps are defined, they are correct as in P and M, but they are not defined for the boundary states (x_1, x_2) where both x_1 and x_2 are boundary states of $\mathcal{A}_1$ and $\mathcal{A}_2$. Thus, except for the double boundary states, B is a proper CPDP element of $\mathcal{A}$.

Finally, we have to check that C is a proper CPDP choice function. Take any $l = (l_1, l_2) \in L$ and any $b : (l_1, l_2) \to (l_1', l_2') \in B$. The CPDP conditions on C are: $C(b, l, (x_1, x_2)) \in [0, 1]$ for all $(x_1, x_2) \in \partial Inv(l_1, l_2)$ and $\sum_{b:(l_1,l_2)\to} C(b, l, (x_1, x_2)) = 1$ for all $(x_1, x_2) \in \partial Inv(l_1, l_2)$. From the definition of C we see directly that $C(b, l, (x_1, x_2)) \in [0, 1]$ for all $(x_1, x_2) \in \partial Inv(l_1, l_2)$ except $\{(x_1, x_2) | x_1 \in \partial Inv_1(l_1)$ and $x_2 \in \partial Inv_2(l_2)\}$. For the second condition, we first look at the case where $x_1 \in \partial Inv_1(l_1)$ and $x_2 \in Inv_2(l_2)$. From rules **r1** and **r2** we can derive that every $b : (l_1, l_2) \to (l_1', l_2') \in B$ corresponds with a unique $b_1 : l_1 \to l_1' \in B_1$ and vice versa. Then from the definition of C we can derive that $C(b, l, (x_1, x_2)) = C_1(b_1, l_1, x_1)$ (**c1**) and consequently

$$\sum_{b:(l_1,l_2)\to} C(b, l, (x_1, x_2)) = \sum_{b_1:l_1\to} C_1(b_1, l_1, x_1) = 1.$$

The case where $x_1 \in Inv_1(l_1)$ and $x_2 \in \partial Inv_2(l_2)$ is symmetrical to the former case and with the same kind of reasoning we find $\sum_{b:(l_1,l_2)\to} C(b, l, (x_1, x_2)) = 1$. For the case where $x_1 \in \partial Inv_1(l_1)$ and $x_2 \in \partial Inv_2(l_2)$ we find that $C(b, l, (x_1, x_2))$ is undefined. We have seen that for all $(l_1, l_2) \in L$, the CPDP conditions on C are met on $\partial Inv(l_1, l_2)$ except for the case where $x_1 \in \partial Inv_1(l_1)$ and $x_2 \in \partial Inv_2(l_2)$. In that case C is undefined, therefore C is a proper CPDP element of $\mathcal{A}$ except for the double boundary states. We resume that $\mathcal{A}$ has seven proper CPDP elements and two CPDP elements that are proper except for the double boundary states. ∎

4. EXAMPLE OF A COMPOSED CPDP

We give an example of the composition of two CPDPs.

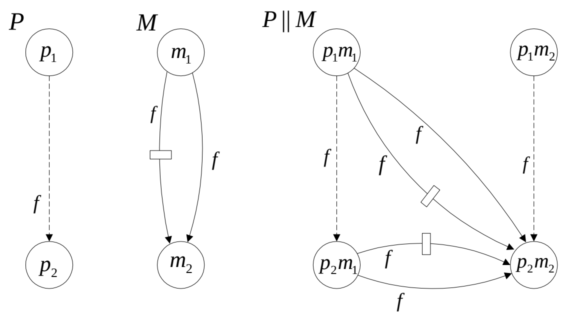

Fig. 1. Two CPDP automata and their composition

In Fig 1 we see the graphical representation of three CPDPs, $\mathcal{P}$, $\mathcal{M}$ and the composition $\mathcal{P}||\mathcal{M}$. Boundary-hit, passive and Markov transitions are respectively pictured as solid arrows, dashed arrows and solid arrows with a little box in the middle. This graphical representation has its origins in Interactive Markov Chains (Hermanns 1999). We give the following interpretation, which is a simplification of the ATM example from (Everdij and Blom 2000). $\mathcal{P}$ is a process which depends on a machine $\mathcal{M}$. $\mathcal{P}$ has two working-modes, a nominal desired mode (p_1) and a failure mode (p_2). The machine $\mathcal{M}$ also has a nominal (m_1) and a failure (m_2) mode. $\mathcal{M}$ is an autonomous machine which can break down in two ways, via a boundary-hit of the continuous process or via a Poisson process. The interrelation between $\mathcal{P}$ and $\mathcal{M}$ is that $\mathcal{P}$ breaks down whenever $\mathcal{M}$ breaks down. This is a one-way synchronization which is reflected by the passive transition.

Now we give the formal description of $\mathcal{P}$, $\mathcal{M}$ and $\mathcal{P}||\mathcal{M}$ where we omit the specification of the continuous processes for simplicity reasons.

$\mathcal{P} = (L_P,\ d_P,\ Inv_P,\ A,\ C_P,\ B_P,\ M_P,\ P_P,\ G_P)$, where $L_P = \{p_1, p_2\}$, $A = \{f\}$, $B_P = M_P = \emptyset$, $P_P = \{(p_1, f, p_2, R_P)\}$ with R_P some reset map.

$\mathcal{M} = (L_M, d_M, Inv_M, A, C_M, B_M, M_M, P_M, G_M)$ where $L_M = \{m_1, m_2\}$, $B_M = \{(m_1, f, m_2, R_{M,1})\}$, $M_P = \{(m_1, f, m_2, R_{M,2}, \lambda_M)\}$, $P_P = \emptyset$. $R_{M,1}$ and $R_{M,2}$ are proper reset maps and λ_M a proper jump-rate-function.

According to the definition of $||$ we find $\mathcal{P}||\mathcal{M} = (L, d, Inv, A, C, B, M, P, G)$ where $L = \{(p_1, m_1), (p_1, m_2), (p_2, m_1), (p_2, m_2)\}$, $B = \{b_{t,1}, b_{t,2}\}$, $M = \{m_{t,1}, m_{t,2}\}$ and $P = \{p_{t,1}, p_{t,2}\}$. From the composition rules **r1,r1'** till **r6,r6'** we find

$$b_{t,1} = ((p_1, m_1), f, (p_2, m_2), R_{b,1})\ (\mathbf{r2'}),$$
$$b_{t,2} = ((p_2, m_1), f, (p_2, m_2), R_{b,2})\ (\mathbf{r1'}),$$
$$m_{t,1} = ((p_1, m_1), f, (p_2, m_2), R_{m,1}, \lambda_1)\ (\mathbf{r4'}),$$
$$m_{t,2} = ((p_2, m_1), f, (p_2, m_2), R_{m,2}, \lambda_2)\ (\mathbf{r3'}),$$
$$p_{t,1} = ((p_1, m_1), f, (p_2, m_1), R_{p,1})\ (\mathbf{r5}),$$
$$p_{t,2} = ((p_1, m_2), f, (p_2, m_2), R_{p,2})\ (\mathbf{r5}),$$

where all reset-maps and jump-rate-functions follow from the rules.

We make some remarks according to this example. Since $\mathcal{P}$ has no boundary-hit transitions, $\mathcal{P}$ has no boundary points at all because a CPDP has at least one boundary-hit transition active for each boundary point. This means that $\mathcal{P}$ and $\mathcal{M}$ can not reach their boundaries at the same time and since this is the only case of underspecification, $\mathcal{P}||\mathcal{C}$ is not underspecified and is therefore a (fully specified) CPDP.

The second remark is on initial states. We did not use the concept of initial states in the definition of CPDPs and therefore any hybrid state is potentially an initial state. In the example above, it might be desirable to indicate location $p_1 \in L_P$ and $m_1 \in L_M$ as initial locations. In that case location (p_1, m_1) would be the initial location of $\mathcal{P}||\mathcal{M}$ and location (p_1, m_2) and its outgoing passive transition would be useless. In a CPDP definition with initial state(s), the definition of $||$ could then be adjusted in such a way that these "useless" or unreachable locations and transitions are discarded.

The last remark is on other composition contexts. The composed system $\mathcal{P}||\mathcal{M}$ still has passive transitions which can be used for interaction in other composition contexts. There can for example be a second autonomous machine $\mathcal{M}_2$ which executes the label f and can therefore bring the process $\mathcal{P}$ into failure mode. If we compose $\mathcal{P}||\mathcal{M}$ in this new context with $\mathcal{M}_2$, we would get the process $(\mathcal{P}||\mathcal{M})||\mathcal{M}_2$. It is intuitively clear that a good composition operator should be commutative and associative, i.e. $\mathcal{P}||\mathcal{M}$ should be equivalent to $\mathcal{M}||\mathcal{P}$ and $(\mathcal{P}||\mathcal{M})||\mathcal{M}_2$ should be equivalent to $\mathcal{P}||(\mathcal{M}||\mathcal{M}_2)$. We are convinced that $||$ satisfies these properties, but it still has to be proven.

5. CONCLUSIONS

In this paper we introduced CPDP (Communicating Piecewise Deterministic Markov Processes) as an automata formalism for compositional specification of PDP systems. A CPDP corresponds to a specific hybrid stochastic process. It was proven in (Strubbe 2003) that CPDPs generate the same class of stochastic processes as PDPs. In this sense CPDPs and PDPs are equivalent.

We introduced the concept of passive transitions to establish a one-way synchronization between systems. We think that the idea of one-way synchronization is very natural in many cases, like the cases where one system is depending on a second system, while the second system is autonomous and therefore not depending on the first system.

This idea of one-way synchronization is formalized in the composition operator that we defined. We proved that the composition of two CPDPs is again a CPDP which might be underspecified. This underspecification however, can not be seen as a shortcoming of the operator, it naturally follows from the fact that a parallel execution of two CPDPs brings forth cases where a choice has to be made. This choice can be made by the composition operator (in that case the composition would result in a CPDP) but we think that it is better to leave this choice open because it might depend on the situation or the kind of application which choice is most suitable.

Communication as we introduced it, takes place via the discrete transitions. In systems theory however, interconnection takes place via the continuous variables. In (Julius *et al.* 2003), the HBA (Hybrid Behavioral Automata) framework is introduced, where both discrete communication via passive transitions and continuous interconnection are present. We think that extending the CPDP framework in the direction of continuous interconnection is an interesting direction for future research.

REFERENCES

Alur, R., C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis and S. Yovine (1995). The algorithmic analysis of hybrid systems. *Theoretical Computer Science* **138**(1), 3–34.

Davis, M.H.A. (1984). Piecewise-deterministic markov processes: A general class of non-diffusion stochastic models. *Journal of the Roy. Stat. Soc. Series B* **46**(3), 353–388.

Everdij, M.H.C. and H.A.P. Blom (2000). Piecewise deterministic markov processes represented by dynamically coloured petri nets. Technical Report NLR-TP-2000-428. National Aerospace Laboratory NLR. Amsterdam.

Hermanns, H. (1999). Interactive Markov Chains. PhD thesis. Universität Erlangen-Nürnberg.

Julius, A.A., S.N. Strubbe and A.J. van der Schaft (2003). Control of hybrid behavioral automata by interconnection. *To appear in the Proceedings of ADHS 2003.*

Lynch, N.A., R. Segala and F.W. Vaandrager (2001). Hybrid I/O automata revisited. In: *Proceedings 4th Int'l Workshop on Hybrid Systems : Computation and Control.* Springer-Verlag. pp. 403–417.

Strubbe, S.N. (2003). Equivalence of pdps and closed cpdps. *Unpublished notes.*

van der Schaft, A.J. and J.M. Schumacher (2001). Compositionality issues in discrete, continuous, and hybrid systems. *Int. J. Robust Nonlinear Control* **11**(5), 417–434.