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

Algorithmic bisimulation for Communicating Piecewise Deterministic Markov Processes

Stefan Strubbe and Arjan van der Schaft

Abstract— **In this paper we present an algorithm for finding a bisimulation relation for stochastic hybrid systems from the class of CPDPs (Communicating Piecewise Deterministic Markov Processes). We prove that the fixed point of the algorithm forms a bisimulation on the state space of the CPDP. We give sufficient conditions on the continuous dynamics and the transition structure of a CPDP, for the computation of the algorithm to be decidable.**

I. Introduction

The class of Communicating Piecewise Deterministic Markov Processes (CPDPs) is an automata framework developped for compositional modelling and analysis of complex stochastic hybrid systems. In [12] and [14] it is shown how complex CPDPs can be modelled in a compositional way by first specifying all component-CPDPs of the complex system and second by interconnecting these component-CPDPs by using a parallel composition operator. In [13] it is shown that the behavior of a CPDP that is closed (i.e. that does not interact with any other systems) can be modelled by a Piecewise Deterministic Markov Process (PDP) (see [2] and [3]). Analysis tools have been developed for PDPs (see [3] and with the equivalence result of [13], these tools can be used for closed CPDPs as well.

It is well-known that the composition of multiple subsystems leads to state space explosion and this is also the case for composite CPDPs. One tool that has proved to be effective in dealing with the state space explosion problem is bisimulation. Bisimulation can be seen as a state space reduction technique: by using bisimulation we can find systems with smaller state spaces, that still have the same external behavior. Two systems have the same external behavior if they cannot be distinguished in any composition context. The notion of bisimulation was introduced by Milner [7] in the context of discrete state processes. Bisimulation has also been established in the context of probabilistic and stochastic automata [6], [1], continuous time interactive Markov chains (IMC) [4], continuous dynamical systems [8], [10], general (non-stochastic) hybrid systems [5], [11], and CPDPs [14].

In this paper we present an algorithm to compute a bisimulation relation on the set of locations L of any CPDP X. In general the computation of this algorithm will not be decidable. We give sufficient conditions on the continuous dynamics, the guards, the reset maps and the jump rates, under which the algorithm terminates (i.e. has a fixed point within finitely many steps) and under which the computation is decidable. One of these conditions is that the number of different probability measures used for the resets of the transitions is finite. For the so called identity-reset map, which plays an important role in the composition of CPDPs, an infinite number of probability measures is needed. We show how the algorithm can be adjusted such that identity-reset maps are allowed, while keeping decidable computation of the algorithm.

The organization of the paper is as follows. In Section II we give definitions of the CPDP model and bisimulation for CPDPs. In Section III we give the algorithm and prove that it provides the maximal bisimulation. In Section IV we give suffucient conditions for the algorithm to be decidable. In Section V we give an example of using the algorithm. Finally, in Section VI we draw conclusions.

II. CPDPs

In this section we define the CPDP-model, we introduce some notation and we define the concept of bisimulation.

A. The CPDP model

We give the formal definition of CPDP as an automaton.

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. $w=G(l,w)$).

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

- 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.

We now introduce some notation. We call an active transition with event $a \in \Sigma$ an a-transition and we call a passive transition with event $\bar{a} \in \bar{\Sigma}$ a $\bar{a}$-transition. 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 $vs(l) := \{(v_1 = r_1, v_2 = r_2, \cdots, v_m = r_m) | r_i \in \mathbb{R}^{d(v_i)}\}$, where v_1 till v_m are the variables from $\nu(l)$, the valuation space or state space of location l and each $(v_1 = r_1, \cdots, v_m = r_m)$ is called a valuation or state of l. We call $\{(l, x) | l \in L, x \in vs(l)\}$ the (hybrid) state space of the CPDP with location set L and valuation spaces $vs(l)$. With the output variables (instead of state variables) we define in the same way output valuations, output space of location l and the (hybrid) output space of the CPDP. If a state x lies in the guard G_α of active transition α, then we say that α is enabled at x. We say that a passive transition α is enabled at x if l_x, the location of x, is the origin location of α. We say that a transition leaves location l if l is the origin location of that transition.

A reset map R of a CPDP consists of an indexed set of probability measures (i.e. R assigns to each state x of a location l a probability measure). We call the probability measures of a reset map *reset measures*. A reset map/measure resets the state variables of a specific location. We call this specific location the *target location* of the reset map/measure.

We can assign a scheduler S_X to any CPDP X. A scheduler is a mechanism that probabilistically chooses which transition is taken given that a transition has to be executed from some hybrid state x. Formally, S_X assigns to each guarded hybrid state x (i.e. each hybrid state that lies in the guard of some active transition) combined with each active transition α of X a value in $[0,1]$ (i.e. $S_X(x,\alpha) \in [0,1]$) such that

$$\sum_{\alpha \in l_x \rightarrow} S(x, \alpha) = 1,$$

where $\alpha \in \mathscr{A}_{l_x}$ is the set of all active transitions that are enabled at x. In this way S_X defines for each x a probability measure on the set of active transitions that are enabled at x. Also, S_X assigns to each hybrid state x combined with each passive transition α a value in $[0,1]$, such that for each $\bar{\alpha} \in \bar{\Sigma}$ we have

$$\sum_{\alpha \in \mathscr{P}_{l_x, \bar{a}}} S(x, \alpha) = 1,$$

where $\alpha \in \mathscr{P}_{l_x, \bar{a}}$ is the set of all $\bar{a}$-transitions that are enabled at x. Thus, S_X also defines for each x and each $\bar{a}$ a probability measure on the set of $\bar{a}$-transitions enabled at x (unless there are no $\bar{a}$-transitions enabled at x).

B. Bisimulation for CPDPs

In order to define bisimulation for CPDPs we need to introduce the notions of *combined reset map* and *combined jump rate function*. We consider CPDP $X = (L, V, W, \nu, w, F, G, \Sigma, \mathscr{A}, \mathscr{P}, \mathscr{S})$, with hybrid state space E, together with scheduler S_X. We define R, which we call the *combined reset map*, as follows. R assigns to each triplet (l, x, a) with $(l, x) \in E$ and with $a \in \Sigma$ and each Borel set $B \in \mathscr{B}(E)$, where $\mathscr{B}(E)$ denotes the set of Borel sets of E, a value in $[0,1]$ (i.e. $R(l,x,a)(B) \in [0,1]$) as follows: for any l' and any Borel set $A \subset vs(l')$

$$R(l, x, a)(l', A) = \sum_{\alpha \in \mathscr{A}_{l,a,l'}} S_X(l, x)(\alpha) R_\alpha(A, x),$$

where $\mathscr{A}_{l,a,l'}$ denotes the set of active a-transitions from l to l' and (l', A) denotes the Borel set $\{(l', x) | x \in A\}$. (This measure is uniquely extended to all Borel sets of E). Now, for $A \in \mathscr{B}(E)$, $R(l,x,a)(A)$ equals the probability of jumping into A via an active transition with label a given that the jump takes place at (l, x).

Furthermore, R assigns to each triplet $(l, x, \bar{a})$, with $(l, x) \in E$ and with $\bar{a} \in \bar{\Sigma}$, and each Borel set $B \in \mathscr{B}(E)$ a value in $[0,1]$ as follows: for any l' and any Borel set $A \subset vs(l')$

$$R(l, x, \bar{a})(l'_1, A) = \sum_{\alpha \in \mathscr{P}_{l,\bar{a},l'}} S_X(l, x)(\alpha) R_\alpha(A, x).$$

(This measure is uniquely extended to all Borel sets of E). Now, $R(l, x, \bar{a})(A)$, with $A \in \mathscr{B}(E)$, equals the probability of jumping into A if a passive transition with label $\bar{a}$ takes place at (l, x).

We define the *combined jump rate function* λ for CPDP X as

$$\lambda(l, x) = \sum_{\alpha \in \mathscr{S}_l} \lambda_\alpha(l, x),$$

with $(l, x) \in E$.

Finally, for spontaneous jumps, R assigns to each $(l, x) \in E$ and each Borel set $B \in \mathscr{B}(E)$ a value in $[0,1]$ as follows: for any l' and any Borel set $A \subset vs(l')$

$$R(l, x)(l'_1, A) = \sum_{\alpha \in \mathscr{S}_{l \rightarrow l'}} \frac{\lambda_\alpha(l, x)}{\lambda(l, x)} R_\alpha(A, x).$$

(This measure is uniquely extended to all Borel sets of E).

The definition of bisimulation for CPDPs, which we are about to present, uses the concepts of *measurable relation* and *equivalent measure*. These notions are formally defined in [14]. Briefly said, a measurable relation $\mathscr{R} \subset X \times Y$ on measurable spaces $(X, \mathscr{B}(X))$ and $(Y, \mathscr{B}(Y))$ is a relation such that the $\mathscr{R}$-projection of each Borel set B of X on Y is a Borel set of Y and vice versa. Then, if $B_X \in \mathscr{B}(X)$ is projected on $B_Y \in \mathscr{B}(Y)$ (and vice versa), B_X and B_Y are each others *corresponding Borel sets*. Two measures r_X and r_Y on X and Y are called equivalent if $r_X(B_X) = r_Y(B_Y)$ for sets B_X and B_Y that correspond to each other.

Definition 2.2: Suppose we have CPDPs $X = (L_X, V_X, W, v_X, w_X, F_X, G_X, \Sigma, A_X, P_X, S_X)$ and $Y = (L_Y, V_Y, W, v_Y, w_Y, F_Y, G_Y, \Sigma, A_Y, P_Y, S_Y)$ with shared W and Σ and with schedulers S_X and S_Y. A measurable relation $\mathscr{R} \subset vs(X) \times vs(Y)$ is a bisimulation if $((l_1, x), (l_2, y)) \in \mathscr{R}$ implies that

1) $\omega_X(l_1) = \omega_Y(l_2)$, for all $w \in \omega_X(l_1)$ we have $G_X(l_1, x, w) = G_Y(l_2, y, w)$, $\lambda(l_1, x) = \lambda(l_2, y)$ (with λ the combined jump rate function defined on both $vs(X)$ and $vs(Y)$).
2) $(\phi_{l_1}(t, x), \phi_{l_2}(t, y)) \in \mathscr{R}$ (with $\phi_l(t, z)$ the state at time t when the state equals z at time zero).
3) If $\lambda(l_1, x) = \lambda(l_2, y) \neq 0$, then $R(l_1, x)$ and $R(l_2, y)$ are equivalent probability measures with respect to $\mathscr{R}$.
4) For any $\bar{a} \in \bar{\Sigma}$ we have that either both $l_1 \overset{\bar{a}}{\not\longrightarrow}$ and $l_2 \overset{\bar{a}}{\not\longrightarrow}$ or else $R(l_1, x, \bar{a})$ and $R(l_2, y, \bar{a})$ are equivalent probability measures.
5) For any $a \in \Sigma$ we have that either both $l_1 \overset{a}{\not\longrightarrow}$ and $l_2 \overset{a}{\not\longrightarrow}$ or else $R(l_1, x, a)$ and $R(l_2, y, a)$ are equivalent measures.

X with initial state (l_1, x) and Y with initial state (l_2, y) are bisimilar if $((l_1, x), (l_2, y))$ is contained in some bisimulation.

We call two locations l_X and l_Y of CPDPs X and Y (where X may be equal to Y) with state spaces E_X and E_Y bisimilar if there exists a bisimulation relation $\mathscr{R} \subset E_X \times E_Y$ such that $vs(l_X) = \{x \in vs(l_X) | (\exists y \in vs(l_Y))((x, y) \in \mathscr{R})\}$ and $vs(l_Y) = \{y \in vs(l_Y) | (\exists x \in vs(l_X))((x, y) \in \mathscr{R})\}$.

III. Bisimulation algorithm

Bisimulation algorithms, that check for example bisimilarity of locations l and l', normally have to check whether a a-transition of l has a 'matching' a-transition in l' and vice versa. In the case of CPDP, this is a bit different. Here, an a-transition of l should not have matching a-transition in l', but rather the combined action of scheduler and all a-transitions of l should match the combined action of scheduler and all a-transitions of l.

We assume throughout this section that guards, jump-rates and the assignment of reset measures can all be defined on the output space. This means that we assume that if x and x' have the same output value, then for any guard G, either both x and x' are in G or both x and x' are not in G, for any spontaneous transition the jump-rates for x and x' are the same and finally, any reset map assigns to x and x' the same reset measure. Therefore, under these assumptions, we can say that a guard or a guard area is a part of the output space, etc. For the bisimulation algorithm that we will present, we restrict ourselves to CPDPs that satisfy these assumptions.

A. Algorithm

Let L be the set of locations of X and let M be the set of all reset measures 'used' by X, i.e., let $M := \{r | (\exists l \in L, x \in vs(l), \sigma \in \Sigma \cup \bar{\Sigma})(r = R(l, x, \sigma))\} \cup \{r | (\exists l \in L, x \in vs(l))(r = R(l, x))\}$. Any reset measure from M either reflects the probability measure that is, for some $\sigma \in \Sigma \cup \bar{\Sigma}$, the result of the probabilities assigned by the scheduler to all σ-transitions and the probabilities of all reset measures of those σ-transitions, or reflects the probability measure that is the result of the combination of reset measures of all spontaneous transitions.

The algorithm that we give next, consists of three steps. In the first step, a partition of L is made, such that locations that have bisimilar continuous dynamics are in the same class of the partition. For each l and l' that lie in the same class of the partition, the maximal continuous bisimilation $\mathscr{R}_{l,l'}$ on the state spaces of l and l' is determined, where a continuous bisimulation is a relation on the state space that satisfies items 1 and 2 of Definition 2.2. Two reset measures r and r' are equivalent with respect to the partition of step one if first l_r and $l_{r'}$ (the target locations of the reset measures) lie in the same class and second if r and r' are equivalent with respect to $\mathscr{R}_{l,l'}$. In the second step, a partition of M is made, such that reset measures that are equivalent with respect to the partition of step one, are in the same class of the partition. In step three, the two partitions on L and M of steps one and two, are refined via inductive steps: in each step a class of one of the partitions is refined if two elements of that class can be discriminated as 'not bisimilar'. The fixed point of step three will turn out to be a bisimulation on the set of locations. In the algorithm, we use the following notation: If $\mathscr{R}$ is an equivalence relation on a set X, then $Part(\mathscr{R})$ denotes the corresponding partition, i.e. $Part(\mathscr{R}) = \{C_1, C_2, \cdots, C_n\}$, where C_1 till C_n are the equivalence classes of X with respect to $\mathscr{R}$.

Before we present the algorithm, we need to introduce the partitioning functions P_s and P_j. These are defined as follows. For all $l \in L$, $\sigma \in \Sigma \cup \bar{\Sigma}$ and $C_r \subset M$, $P_s(l, \sigma, C_r) :=$

$$\bigcup_{p \in \mathbb{R}_+} \{(p, \{y | \sum_{\alpha \in \mathscr{T}_{l,y\overset{\sigma}{\rightarrow}}} S(y)(\alpha) = p,$$

$$\sum_{\alpha \in \mathscr{T}_{l,y\overset{\sigma}{\rightarrow}}} S(y, \alpha) R_\alpha(y) \in C_r\} - \bigcup_{p \in \mathbb{R}_+} \{(p, \emptyset)\},$$

where $\mathscr{T}_{l,y\overset{\sigma}{\rightarrow}}$ denotes the set of all σ-transitions leaving l that are enabled at output y, and $R_\alpha(y)$ denotes the reset measure of transition α at output y. For all $l \in L$ and $C_r \subset M$, $P_j(l, C_r) :=$

$$\bigcup_{\lambda \in \mathbb{R}_+} \{(\lambda, \{y | \sum_{\alpha \in \mathscr{S}_{l\rightarrow}} \lambda_\alpha(y) = \lambda,$$

$$\sum_{\alpha \in \mathscr{S}_{l\rightarrow}} \frac{\lambda_\alpha(y)}{\lambda} R_\alpha(y) \in C_r\})\} - \bigcup_{\lambda \in \mathbb{R}_+} \{(\lambda, \emptyset)\}.$$

The functions P_s and P_j have the following interpretations. For $\sigma \in \Sigma$, $P_s(l,\sigma,C_r) = \cup_{i\in I}\{(p_i,S_i)\}$ for some index set I. If $y \in S_i$ for some $i \in I$, then this means that if an active transition is executed at a state with output y, then 1. the probability that that the scheduler chooses a σ-transition equals p_i and 2. the reset map that is used then, lies in C_r. Note that $\cup_{i\in I}S_i$ forms a partition of $\{y|R(l,y,\sigma) \in C_r\}$. For $\sigma \in \bar{\Sigma}$ we get $P_s(l,\sigma,C_r) = \{(1,S)\}$. Here, S is the set of outputs where the combined action of scheduler and all σ-transitions results in a reset measure from C_r. $P_j(l,C_r) = \cup_{i\in I}\{(\lambda_i,S_i)\}$ for some index set I. For $y \in S_i$, this means that 1. the total jump-rate for this output y equals λ_i and 2. the total reset map of the spontaneous transitions for output y lies in C_r.

Algorithm 1:

Step 1.Determine $\mathscr{R}_l^0 \subset L \times L$ such that $(l,l') \in \mathscr{R}_l^0$ if and only if the continuous dynamics of l and l' are bisimilar. Then, determine for each $(l,l') \in \mathscr{R}_l^0$ the maximal bisimulation concerning the continuous dynamics $\mathscr{R}_{l,l'} \subset vs(l) \times vs(l')$.

Step 2.Determine $\mathscr{R}_r^0 \subset R \times R$ such that $(r,r') \in \mathscr{R}_r^0$ if and only if r and r' are equivalent with respect to $\mathscr{R}^0 := \{((l_1,x_1),(l_2,x_2))|(l_1,l_2) \in \mathscr{R}_l^0,\ (x_1,x_2) \in \mathscr{R}_{l_1,l_2}\}$.

Step 3.Determine inductively for each $k \in \mathbb{N}$

$$\mathscr{R}_l^{k+1} = \mathscr{R}_l^k \cap \{(l,l')|(\forall\sigma \in (\Sigma \cup \bar{\Sigma}))(\forall C_r \in \mathscr{R}_r^k)$$
$$(P_s(l,\sigma,C_r) = P_s(l',\sigma,C_r) \wedge P_j(l,C_r) = P_j(l',C_r))\}$$
$$\mathscr{R}_r^{k+1} = \mathscr{R}_r^k \cap \{(r,r')|(\forall C_l \in \mathscr{R}_l^{k+1})(r \sim r')\},$$

where $r \sim r'$ means that r and r' are equivalent with respect to $\mathscr{R}^{k+1} := \{((l_1,x_1),(l_2,x_2))|(l_1,l_2) \in \mathscr{R}_l^{k+1}, (x_1,x_2) \in \mathscr{R}_{l_1,l_2}\}$.

Theorem 3.1: If algorithm 1 has a fixed point $\mathscr{R}_l^k$ for some $k \in \mathbb{N}$, then this fixed point is a bisimulation on the set of locations.

Proof: We prove that, according to Definition 2.2, $\mathscr{R}^k$ is a bisimulation. Suppose $((l_1,x_1),(l_2,x_2)) \in \mathscr{R}^k$, then

1. $\omega(l_1) = \omega(l_2)$ and $G(l_1,x_1,w) = G(l_2,x_2,w)$ follow from $(l_1,l_2) \in \mathscr{R}_l^0$. If we define for all $C_r \in Part(\mathscr{R}_r^k)$

$$\lambda_{C_r}(x) = \begin{cases} \tilde{\lambda} & (\exists(\tilde{\lambda},G) \in P_j(l,C_r))(x \in G) \\ 0 & \text{otherwise.} \end{cases},$$

then we can write $\lambda(x) = \sum_{C_r \in Part(\mathscr{R}_r^k)} \lambda_{C_r}(x)$ and then $\lambda(x_1) = \lambda(x_2)$ follows from $P_j(l_1,C_r) = P_j(l_2,C_r)$.

2. Follows from $(l_1,l_2) \in \mathscr{R}_l^0$.

3. Take any $C_l \in Part(\mathscr{R}_l^k)$ and any saturated Borel set B within the state space of C_l. Let $B_{/\mathscr{R}^k}$ denote the projection of B on the quotient hybrid state space (with respect to $\mathscr{R}^k$) and let r_{C_r} denote the reset map on the level of this quotient space corresponding to the (equivalent) reset maps in C_r. Then it can be seen that

$$R(l_1,x_1)(B) = \sum_{C_r \rightarrow C_l} \frac{\lambda_{C_r}(x_1)}{\lambda(x_1)} r_{C_r}(B_{/\mathscr{R}^k}),$$

where $C_r \rightarrow C_l$ denotes $C_r \in \{C \in \mathscr{R}_r^k|$ the target location of any $r \in C$ lies in $C_l\}$, and because $\lambda_{C_r}(x_1) = \lambda_{C_r}(x_2)$ we get $R(l_1,x_1)(B) = R(l_2,x_2)(B)$. It can now be easily seen that this result is also valid for any saturated Borel set of the hybrid state space.

4. This can be proved analogously to 3. except that here we get for $\bar{a} \in \bar{\Sigma}$ that

$$R(l_1,x_1,\bar{a})(B) = \sum_{C_r \rightarrow C_l} \frac{p_{C_r}(x_1,\bar{a})}{\lambda(x_1)} r_{C_r}(B_{/\mathscr{R}^k}),$$

where

$$p_{C_r}(x,\bar{a}) := \begin{cases} \tilde{p} & (\exists(\tilde{p},G) \in P_s(l,\bar{a},C_r))(x \in G) \\ 0 & \text{otherwise.} \end{cases}$$

Then because $P_s(l_1,\bar{a},C_r) = P_s(l_2,\bar{a},C_r)$ we get $p_{C_r}(x_1,\bar{a}) = p_{C_r}(x_2,\bar{a})$ and consequently $R(l_1,x_1,\bar{a})(B) = R(l_2,x_2,\bar{a})(B)$.

5. Analogous to 4. except that we use $a \in \Sigma$ instead of $\bar{a} \in \bar{\Sigma}$.

All conditions of Definition 2.2 are satisfied, which means that $\mathscr{R}^k$ is a bisimulation.

■

Remark 3.2: We can not claim that the fixed point of algorithm 1 is the maximal bisimulation on the set of locations. This is because there are situations possible where a class of locations is refined by the algorithm while all locations in that class are bisimilar. This happens when two locations jump to bisimilar states of non-bisimilar locations. These locations do not fall in the same class by the algorithm because they have reset measures to non-bisimilar locations. However, the locations might be bisimilar.

IV. Decidability

Algorithm 1 is a general algorithm for CPDP bisimulation. It will not be decidable in general. In this section we pose some conditions under which the algorithm terminates at a fixed point and is decidable. One of the conditions that we state for decidability is that the set of reset measures (used by the CPDP) is finite. From a compositionality point of view, the identity reset map (i.e. the reset map that leaves the state variables unaltered) is very important. However, because each continuous state has its own identity reset measure, the number of reset measures used by the identity reset map is infinite. At the end of this section we provide a method which, under certain decidability conditions, finds the fixed point of the algorithm while allowing the use of identity reset maps.

A. Decidability conditions

Each of the three steps of Algorithm 1, asks for its own decidability conditions. Because of the lack of space, we refer to [15] for a more extensive description of the decidability conditions. Here we only summarize the results.

Step 1. This step is decidable if the continuous state/output dynamics is for each location linear, i.e. can be described as $\dot{x} = Ax, y = Cx$. Finding the maximal bisimulation between two linear state/output systems can be done by using the method of [10].

Step 2. This step is decidable if M, the set of used reset measures is finite and if each reset measure consists of assigning a positive probability to a finite number of states.

Step 3. This step is decidable if

- the guard of each transition can be given as a finite set of linear inequalities and
- the number of different scheduling values is for each transition finite and the 'guard-areas of equal scheduling value' can all be given as a finite set of linear inequalities and
- for each transition, the number of reset measures used by the reset map of the transition is finite and the guard area corresponding to each reset measure can be given as a finite set of linear inequalities and
- for each spontaneous transition the number of different jump rates is finite and the area corresponding to each jump-rate can be given as a finite set of linear inequalities.

Then for each $a \in (\Sigma \cup \bar{\Sigma})$ and each finite set of reset measures C_r, the computation of $P_s(l,a,C_r)$ and of $P_j(l,C_r)$ is decidable. If then the number of transitions used is finite, then under the conditions listed above, the computation of step 3 is decidable.

B. Identity reset maps

The identity reset map for a state variable v is defined as $Id_v(\{(v=x)\},(v=x)) = 1$, with other words if variable v is reset with Id_v at the moment that its value equals x, then the probability that the value of v after reset equals x, equals one (because the singleton Borel set $\{(v=x)\}$ gets measure one). If we have two CPDPs, X_1 and X_2, interacting with each other, and CPDP X_1 executes a transition (with reset map R) which does not influence X_2, then in the composite CPDP the variables of X_1 are reset with R while the variables of X_2 are reset with Id, which expresses that X_2 is not influenced by the transition of X_1. This is a common situation, which makes clear the importance of identity reset maps. Another situation where the identity reset map is used, is when a CPDP component wants to send, at a certain state, a signal a to another component, while the state should not be changed. This can be expressed via an active a-transition with identity reset map.

In order to use identity reset maps, while still allowing decidable computation of Algorithm 1, we need to syntactically add more structure to the relation between state and output variables of a CPDP location. Composing CPDPs naturally leads to different compartments of a joint location (i.e. a location of the composite CPDP), where each compartment contains the state and output variables of a specific component-CPDP. Instead of ν and ω, which select state and output variables for each location, we will now use γ, which assigns a set of compartments to each location. A compartment is combination of a set of state variables and a set of output variables like $(\{v_1,v_2\},\{w_1,w_2\})$ and we might have for example for some location l that $\gamma(l) = \{(\{v_1,v_2\},\{w_1,w_2\}),(\{v_3,v_4\},\{w_3\})$. Two compartments of one location have disjoint sets of state variables and have disjoint sets of output variables. Output variables may depend only on the state variables of its compartment.

Now we define the concept of extended reset measure: An extended reset measure on a location l assigns to each compartment of location l either a reset measure on the state variables of that compartment, or the symbol Id. Assigning the symbol Id to a compartment, expresses that this compartment is reset via the identity reset.

For Algorithm 1, instead of using the set of all reset measures used by the CPDP we now use the set of all extended reset measures used by the CPDP. Note that under the conditions stated in Section IV-A, this set of extended reset measures is finite, while we can still express he identity reset action. We call extended reset measures r_1 and r_2 *equivalent* if first for each compartment with output set W that is reset by r_1 there is a compartment with output set W that is reset by r_2 and vice versa, and second, two corresponding compartments should be reset by equivalent reset measures of r_1 and r_2 or should both get the symbol Id from r_1 and r_2. Now it can be seen that Algorithm 1, where now M is the set of extended reset measures and the phrase 'equivalent measures' in step 2 is changed to 'equivalent extended measures', determines a bisimulation for CPDPs with compartments and extended reset measures.

V. Example

We give an example of finding the a bisimulation for a CPDP by using Algorithm 1. Consider the CPDP X in Figure 1. X has three locations with state variables x_1,x_2 and x_3 respectively. All locations have linear continuous dynamics and share the same output variable y. There are two spontaneous transitions with label $\lambda[y<0]$ which is shorthand notation and means that the jump-rates for these transitions are λ for $y<0$ and are zero for $y \geq 0$. The other spontaneous transitions have constant rates (μ or 2μ). Furthermore, X has a number of transitions with label a, which stand for active a-transitions with guards equal to the output space $y \in \mathbb{R}$. We assume that CPDP X uses only three reset measures: r_1, r_2 and r_3, where r_i resets the state x_i (of location l_i) deterministically to $\hat{x}_i$. $\hat{x}_i$ is some given 'initial' state for location l_i. We assume that the reset map of any transition of X to location l_i uses (for all $y \in \mathbb{R}$) reset measure r_i. We assume that the scheduler S_X is defined on the five a-transitions as follows:

- $S_X((l_2,y),a,l_2 \xrightarrow{a} l_1)$ equals 1 for $y \leq 0$ and equals 0 for $y > 0$,
- $S_X((l_2,y),a,l_2 \xrightarrow{a} l_3)$ equals 0 for $y \leq 0$ and equals 1 for $y > 0$,
- $S_X((l_3,y),a,l_3 \xrightarrow{a} l_1)$ equals 1 for $y \leq 0$ and equals 0 for $y > 0$,
- $S_X((l_3,y),a,l_3 \xrightarrow{a} l_2)$ equals 0 for $y \leq 0$ and equals $\frac{1}{2}$ for $y > 0$,
- $S_X((l_3,y),a,l_3 \xrightarrow{a} l_3)$ equals 0 for $y \leq 0$ and equals $\frac{1}{2}$ for $y > 0$.

Now we execute Algorithm 1 for CPDP X.

Step 1: Via the algorithm of [10], we can find via matrix operations maximal bisimulations for the continuous

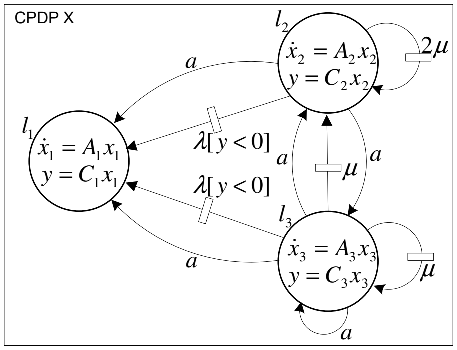

Fig. 1. Example of a CPDP

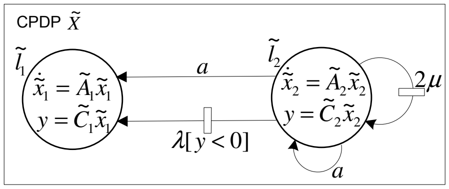
CPDP $\tilde{X}$
$\tilde{l}_1$
$\dot{\tilde{x}}_1 = \tilde{A}_1 \tilde{x}_1$
$y = \tilde{C}_1 \tilde{x}_1$
$\tilde{l}_2$
$\dot{\tilde{x}}_2 = \tilde{A}_2 \tilde{x}_2$
$y = \tilde{C}_2 \tilde{x}_2$
a
$\lambda[y<0]$
2μ
a

Fig. 2. Example of a bisimulation-reduced CPDP

dynamics of locations l_1,l_2 and l_3. We assume that according to this algorithm all three locations are bisimilar, thus we get $\mathscr{R}_c = \{(l_i,l_j)|i,j \in \{1,2,3\}\}$ and $\mathscr{R}_l^0 = \{\{l_1,l_2,l_3\}\}$. Furthermore we assume that maximal state reduced versions of the dynamics of l_1,$l_2$and l_3, which can be computed with the same algorithm, are given by $\dot{\tilde{x}}_1 = \tilde{A}_1\tilde{x}_1$, $y = \tilde{C}_1\tilde{x}_1$ and $\dot{\tilde{x}}_2 = \tilde{A}_2\tilde{x}_2$, $y = \tilde{C}_2\tilde{x}_2$ and $\dot{\tilde{x}}_3 = \tilde{A}_3\tilde{x}_3$, $y = \tilde{C}_3\tilde{x}_3$. We also assume that, according to these computed bisimulations, the states $\hat{x}_1$,$\hat{x}_2$ and $\hat{x}_3$ are bisimilar to one another.

Step 2: From the results and assumptions of step 1 above, it is clear that we get $\mathscr{R}_l^0 = \{(r_i,r_j)|i,j \in \{1,2,3\}\}$ and $\mathscr{R}_r^0 = \{\{r_1,r_2,r_3\}\}$.

Step 3: We can compute that for l_1 we get $P_s(l_1,a,\{r_1,r_2,r_3\}) = 0$, $P_j(l_1,\{r_1,r_2,r_3\}) = 0$, for l_2 we get $P_s(l_2,a,\{r_1,r_2,r_3\}) = \{(1,y \in \mathbb{R})\}$, $P_j(l_2,\{r_1,r_2,r_3\}) = \{(\lambda + 2\mu, y < 0),(2\mu, y \geq 0)\}$ and for l_3 we get $P_s(l_3,a,\{r_1,r_2,r_3\}) = \{(1,y \in \mathbb{R})\}$, $P_j(l_3,\{r_1,r_2,r_3\}) = \{(\lambda + 2\mu, y < 0),(2\mu, y \geq 0)\}$. This means that we get $\mathscr{R}_l^1 = \{\{l_1\},\{l_2,l_3\}\}$ and $\mathscr{R}_r^1 = \{\{r_1\},\{r_2,r_3\}\}$.

We continue with these new partitions and compute that for l_2 we get $P_s(l_2,a,\{r_1\}) = \{(1,y \leq 0)\}$, $P_s(l_2,a,\{r_2,r_3\}) = \{(1,y > 0)\}$, $P_j(l_2,\{r_1\}) = \{(\lambda,y < 0)\}$, $P_j(l_2,\{r_2,r_3\}) = \{(2\mu,y \geq 0)\}$ and for l_3 we get $P_s(l_3,a,\{r_1\}) = \{(1,y \leq 0)\}$, $P_s(l_3,a,\{r_2,r_3\}) = \{(1,y > 0)\}$, $P_j(l_3,\{r_1\}) = \{(\lambda,y < 0)\}$, $P_j(l_3,\{r_2,r_3\}) = \{(2\mu,y \geq 0)\}$. This means that we get $\mathscr{R}_l^2 = \{\{l_1\},\{l_2,l_3\}\}$ and $\mathscr{R}_r^2 = \{\{r_1\},\{r_2,r_3\}\}$, which is the fixed point of the algorithm.

It can be seen that, because here we have no situations as described in Remark 3.2, $\mathscr{R}_l^2$ forms the maximal bisimulation on the set of locations.

VI. Conclusions

We presented an algorithm for finding a bisimulation relation on the set of locations of a CPDP. We showed that if the algorithm terminates, then the result equals a bisimulation. We have given conditions on the continuous dynamics, the reset maps and the transitions, under which the algorithm terminates in a finite number of steps and under which the algorithm is thus decidable. We have shown that for CPDPs that use identity reset maps (which use an infinite number of reset measures, but form from a compositionality point of view an important class of reset maps) we can alter the algorithm such that it stays decidable while allowing identity reset maps.

A direction for future research is to find broader classes of CPDP transitions and reset maps (like perhaps reset maps with Gaussian distributions) that do allow decidable algorithms for finding bisimulations. A second direction is to find optimal ways of computing the three steps of Algorithm 1. It might be possible to combine optimization strategies used in for example [4] for Interactive Markov Chains and in [9] for Switched Linear Systems. A third direction is to get more insight into the maximality of the algorithm; in which cases does the algorithm provide a maximal bisimulation?

References

[1] P. R. D'Argenio, "Algebras and automata for timed and stochastic systems," Ph.D. dissertation, University of Twente, 1997.

[2] 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.

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

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

[5] G. Lafferriere, G. Pappas, and S. Sastry, "Hybrid systems with finite bisimulations," *Lecture Notes in Computer Science*, vol. 1567, pp. 186–203, 1999.

[6] K. G. Larsen and A. Skou, "Bisimulation through probabilistic testing," *Information and Computation*, vol. 94, pp. 1–28, 1991.

[7] R. Milner, *Communication and Concurrency*. Prentice Hall, 1989.

[8] G. Pappas, "Bisimilar linear systems," *Automatica*, vol. 39, pp. 2035–2047, 2003.

[9] G. Pola, A. van der Schaft, and M. di Benedetto, "Bisimulation theory for switching linear systems," 2005, Proceedings of the 43st IEEE Conference on Decision and Control, CDC 04, Paradis Island, USA.

[10] A. van der Schaft, "Bisimulation of dynamical systems," in *HSCC 2004*, ser. Lecture Notes in Computer Science, vol. 2993. Springer, 2004, pp. 555–569.

[11] A. van der Schaft, "Equivalence of hybrid dynamical systems," in *Proceedings of the 16th international symposium on Mathematical Theory of Networks and Systems*, 2004.

[12] 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.

[13] S. N. Strubbe and R. Langerak, "A composition operator for complex control systems," 2005, Accepted for 25th IFIP WG 6.1 International Conference on Formal Techniques for Networked and Distributed Systems, Taipei.

[14] 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.

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