Continuous Time and/or Continuous Distributions

Joseph Assouramou* and Josée Desharnais*

Department of Computer Science and Software Engineering
Université Laval, Québec, Canada
joseph.assouramou.1@ulaval.ca, josee.desharnais@ift.ulaval.ca

Abstract. We compare two models of processes involving uncountable space. Labelled Markov processes are probabilistic transition systems that can have uncountably many states, but still make discrete time steps. The probability measures on the state space may have uncountable support. Hybrid processes are a combination of a continuous space process that evolves continuously with time and of a discrete component, such as a controller. Existing extensions of Hybrid processes with probability restrict the probabilistic behavior to the discrete component. We use an example of an aircraft to highlight the differences between the two models and we define a generalization of both that can model all the features of our aircraft example.

1 Introduction

For many years now, two models of continuous processes have evolved independently. Labelled Markov processes (LMPs) [2] are probabilistic transition systems that can have uncountably many states, but still make discrete time steps. Hybrid processes [1,12] are a combination of a discrete component, such as a controller and a physical process that evolves continuously with time. Existing extension of Hybrid processes with probabilities [14] restrict the probabilistic behaviour to the discrete component.

For both models, notions of bisimulation and simulation, logics, model checking techniques and tools have been developed in disjoint research communities. These models face similar challenges, as their main complexity is of course their continuous nature. For example, notions of approximations have been developed on both sides [10,16,7]. No systematic comparison between the two models has been proposed until now. Nevertheless, it is not clear at first sight if the continuous time model could not encompass the discrete time one. Moreover, in order to implement a model-checking tool for LMPs, a finite language had to be defined to describe LMPs. This was done by Richard [13]. The input language to the tool is an LMP whose probability transitions are combinations of known distributions such as Uniform, Normal, etc. One could wonder if this finiteness has some link with the discrete component of hybrid automata.

* Research supported by NSERC.

A. Aldini et al. (Eds.): EPEW 2010, LNCS 6342, pp. 99–114, 2010.

If the two models have evolved independently, is that to say that they are made for different purposes and hence that there is no need for a unifying framework? On another hand, one could wonder whether there is some satisfactory translation from one to the other. In this paper, we answer these questions by way of a case study. We will show how to model with both frameworks a simple process representing an aircraft flying. Though simplified, this aircraft will be interesting enough to highlight the differences and limitations of the two frameworks considered. These observations will lead us to define a generalization of labelled Markov processes and hybrid probabilistic processes, that we will call hybrid LMPs. Hence, this model will combine both distinguishing features of the two frameworks, that is, continuous time and continuous distributions.

1.1 The Aircraft

Our base example is an aircraft taking off and travelling. Initially, the aircraft is on the ground and once started, it rises up as long as the maximum altitude is not reached. The rising rate is between 20 and 25 m by unit of time. At any time, the pilot may rotate to the right or left; while rotating, the airplane, under certain friction forces (air resistance, etc.), may lose some altitude, following a probability law. If it gets to a zero altitude after rotating, we will assume that it has crashed. Moreover if the pilot tries to rotate the aircraft when the altitude is below the minimal value $H_{\min}$, the aircraft crashes. Before formalizing further this case study, we observe that this example describes a probabilistic system with a continuous state space since the altitude is a real value. Moreover, the system is a continuous time system because the height evolves with time. The system allows only one action that we call "*rotate*". To keep the example simple, we do not allow other actions such as decreasing altitude or landing.

In order to get hands on a precise instance of this model, we now choose a probability function that witnesses the behavior described above. We expect that after a rotation, the plane is more likely to jump to some close height than far from its previous height. Thus, the probability that from an altitude s, the aircraft loses about 20 meters of height should be greater than its probability to lose 50 meters. Assume that the possible values of the altitude are in $R := [0, H_{\max}] \cup \{\mathit{Crashed}\}$ where $H_{\max}$ is the maximum altitude. We will denote by $p(s, [a, b])$ the probability that from altitude s, the aircraft's altitude gets in $[a, b]$ after a rotation. We choose an exponential distribution on $s - x$ where s is the altitude of the aircraft when the pilot makes a rotation. This will make sure that intervals of height values that are closer to s will get greater probability. We define p as follows, where s is in units of height (not necessarily meters):

- if $0 \leq s < H_{\min}$, $p(s, \{\mathit{Crashed}\}) := 1$
- if $H_{\min} \leq s \leq H_{\max}$, $p(s, \cdot)$ is the unique probability measure extension of the following set function (the case $a \leq s < b$ can be deduced)
  - $p(s, [a, b]) := \begin{cases} \int_a^b e^{-(s-x)} dx & \text{if } 0 < a \leq b \leq s \\ p(s, [a, s]) & \text{if } s < b \quad \text{hence 0 if } s < a \end{cases}$
  - $p(s, \{\mathit{Crashed}\}) := \int_{-\infty}^{0} e^{-(s-x)} dx = e^{-s}$

Note that this choice of probability measure to model the aircraft does not record the possible instability that may arise from the rotation: we assume that the altitude loss is measured once the aircraft has stabilized. We also assume that only a loss of altitude is possible (i.e. if $s \leq a \leq b$, $p(s, [a, b]) = 0$). This simplification will play no role in the point we want to make about this benchmark. In the two models that we will present, we will need numerical values to express properties such as "at a an altitude greater than $H_{\min}$, the probability that the aircraft loses 100 meters or more when rotating is between 25% and 50%": in these cases, our unit of height will be 100 meters and $H_{\min} := 5$.

In the following section, we recall different definitions; in Section 3, we will present models of the aircraft using the two frameworks together with a few properties that can or cannot be verified faithfully; and finally, in Section 4, we propose a generalization of both frameworks.

2 Background

In this section, we recall some definitions, including those of the two models that we will analyze in this paper.

A *measurable space* is a pair (S, Σ) where S is any set and $\Sigma \subseteq \mathcal{P}(S)$ is a *σ-algebra* over S, that is, a collection of subsets of S containing S and closed under complementation and countable intersection. Well-known examples are $[0, 1]$ and $\mathbb{R}$ equipped with their respective *Borel* σ-algebras $\mathcal{B}$, generated by intervals. Throughout the paper, we assume the Borel σ-algebra on $[0, 1]$ and $\mathbb{R}$ and we write $\mathcal{B}_n$ for the Borel σ-algebra on $\mathbb{R}^n$. A map f between two measurable spaces (S, Σ) and (S', Σ') is said to be *measurable* if for all $A' \in \Sigma'$, $f^{-1}(A') \in \Sigma$. A necessary and sufficient criterion for measurability of a map $f : (S, \Sigma) \to ([0, 1], \mathcal{B})$ is that the condition be satisfied for $A' := [r, 1]$, for any rational r. A *subprobability measure* on (S, Σ) (or probability distribution) is a map $p : \Sigma \to [0, 1]$, such that for any countable collection (E_n) of pairwise disjoint sets, $p(\cup_n E_n) = \sum_n p(E_n)$. We say that p is discrete when its support is finite or countable, that is, if there is a countable set $E = \mathrm{supp}(p) := \{s \in S : p(\{s\}) > 0\} \in \Sigma$ such that $p(E) = p(S)$. The set of discrete distributions over S will be denoted by $\mathrm{Dist}(S)$, and the set of all subprobability measures, $\mathsf{Sub}(S)$.

2.1 Labelled Markov Processes (LMPs)

Labelled Markov Processes are used to model reactive and probabilistic systems whose state spaces might be continuous, that is, systems that react to events in their environment and may have uncountably many states.

Definition 2.11 ([2]). *Let Act be a countable set of actions and AP a countable set of atomic propositions. A* Labelled Markov Process (LMP) *is a tuple $(S, \Sigma, i, \{\mu_a\}_{a \in Act}, Label)$ where:*

- *S the set of* states*, (S, Σ) is a measurable space and $i \in S$ is the initial state*
- *$\mu_a : S \times \Sigma \to [0, 1]$ is a transition (sub)probability function $\forall a \in Act$, i.e.,*

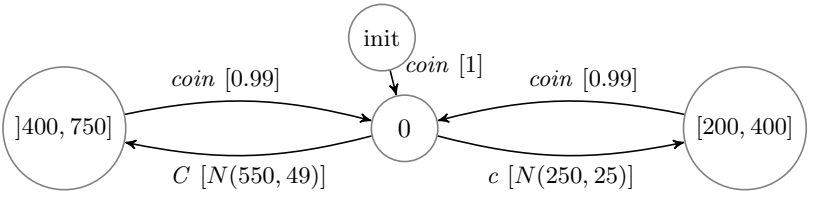

Fig. 1. LMP of a coffee maker 2.12

- *for each* $s \in S$, $\mu_a(s, \cdot)$ *is a subprobability measure and*
- *for each* $E \in \Sigma$, *the function* $\mu_a(\cdot, E)$ *is measurable*

– $Label : S \to \mathcal{P}(AP)$ *is a measurable function used to describe states.*

The value $\mu_a(s, E)$ is the probability that, starting in state s and under the request of action a, the system makes a transition into one of the states in E, in one unit of time; hence time is discrete. To model a continuous time phenomenon using an LMP, one can monitor its values at fixed intervals of time, making the time discrete. An extension of LMPs allowing external non deterministic transitions is obtained essentially by allowing a (countable) set of distributions for every action. For a complete treatment of non deterministic LMPs, one can refer to D'Argenio et al. [6] or Cattani et al. [5]. A typical example of LMP is an interacting machine.

Example 2.12. *Consider a coffee machine with two buttons* c *and* C *for serving a small and a large coffee respectively. Once the coffee is served, in* 99% *of the case, the system is ready to serve a coffee as soon as a coin is inserted. We limit the state space to* $\{\text{init}, 0\} \cup [200, 750]$*: initially, the machine is in state* init *and real numbers represent the amount of coffee served when the customer presses one of the two buttons. The small and the large coffee correspond to states in the intervals* $[200, 400]$ *and* $]400, 750]$ *respectively. We associate a distribution* μ_c *that follows a Normal law of mean 250 and variance 25 to the small coffee button and another Normal distribution* μ_C *of mean 550 and variance* 49 *to the large coffee button. An LMP model of such a system is defined over* $AP := \{small,large\}$*, as* $(\{\text{init}, 0\} \cup [200, 750], \mathcal{B}, 0, \{\mu_c, \mu_C, \mu_{coin}\}, Label)$*, where* $Label([200, 400]) := \{small\}$ *and* $Label(]400, 750]) := \{large\}$*. It is depicted in Figure 1.*

In graphic representations, each transition is labelled with an action followed, in brackets, by the probability distribution that is restricted to the target set. Thus, the transition labelled by $c[N(250, 25)]$*, encodes a* c*-transition from the state* 0 *whose subprobability measure function follows a Normal distribution on the target set* $[200, 750[$*. If the capacity of large glasses is less than* 700 ml*, we can compute the probability that the coffee overflows, which is given by* $\mu_C(0, [700, 750])$*. So, there is* 1% *chance that a coffee overflows.*

Model-Checking Methods. Because of their infinite state space, the model-checking of LMPs can only be done for sub-families of processes that exhibit

some form of regularity. Richard [13] has developed Cismo[1], a model-checking tool to verify next state probabilistic properties (possibly nested and branching). A language for LMPs was defined in order to feed the model-checker: the state spaces allowed are powers of the reals. The input language to the tool is an LMP whose transition probability functions are combinations of known distributions such as Uniform, Normal, or any repartition function.

2.2 Probabilistic Hybrid Systems

Hybrid systems are dynamic systems that combine discrete and continuous changes, and thus, they are defined as the composition of a dynamic continuous system and a discrete one. Contrarily to the LMP model, the change of state is continuous with time. The following (non probabilistic) example of a monitor is typical.

Example 2.21. *Consider a monitor that measures the performance of a cement factory machine and assume that this performance is measured continuously on a scale of* $[0, 1]$*. Thus, when the performance is* 0.5*, the machine is used at* 50% *of its maximum resources. We suppose that there exist three operating machinery modes: "Stable" is the normal mode of operation where the performance of the machine increases linearly in the interval* $[0.8, 1]$*; in mode "Unstable", the performance varies at a rate between* $[-\frac{1}{64}, \frac{1}{64}]$ *in the interval* $[0.4, 0.8]$*; finally, "Low" is the mode in which the machine's performance is smaller than* 0.4 *and decreases gradually at a linear rate. Discrete steps can happen when, depending on the needs, the operator decides to change the machine's mode by pressing on one of the buttons: "low mode", "stable mode", "unstable mode".*

Probabilistic hybrid systems are hybrid systems for which relative likelihoods are associated with certain behavior. Therefore, we can talk about probabilistic timed properties such as "with a probability greater than 0.9, some system will be in some target state within 3 minutes".

Example 2.22. *Consider the monitor of Example 2.21, but now assume that there is only one button to switch from one mode to another. From the mode "Stable", by pressing the button, the machine may switch either to mode "Low", with probability* 0.3 *or to the mode "Unstable", with probability* 0.7*.*

The formalism used for the specification and verification of hybrid systems was introduced independently by Alur and al. [1] and Nicollin and al. [12]. Sproston has extended their formalism with probabilistic behaviour [14,15]. There are many subclasses of hybrid systems, among which linear [10] and rectangular [11]. On the probabilistic side, the same classes are defined; for simplicity, we will restrict to the probabilistic rectangular ones. We first need some notation. Given a finite set X of real variables, we write $\dot{X} := \{\dot{x}_1, ..., \dot{x}_n\}$ where $\dot{x}_i := \frac{dx_i}{dt}$ is the first derivative of x_i with respect to time. A valuation a is a function $\mathsf{a} : X \to \mathbb{R}$ that assigns values to variables; we equivalently write $\mathsf{a} \in \mathbb{R}^n$ where $n = |X|$.

[1] More can be found on the web site of Cismo:
`http://www2.ift.ulaval.ca/~jodesharnais/VPS/`

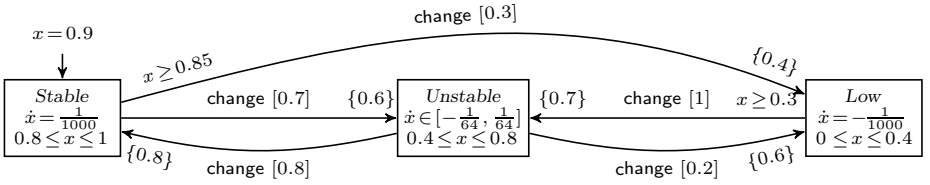

Fig. 2. The PHA of a machine factory's monitor

Definition 2.23. *Let X be a finite set of variables. A set U of valuations is* rectangular *if there exists a family of (possibly unbounded) intervals $(I_x)_{x\in X}$ with rational endpoints such that $U = \{\mathsf{a} \in \mathbb{R}^n \mid \mathsf{a}(x) \in I_x \text{ for all } x \in X\}$. We denote by $R(X)$ the set of rectangles over X.*

Definition 2.24 ([15]). *A* probabilistic rectangular hybrid automaton (PHA) *is a structure $H = (X, V, init, A, inv, flow, prob, \langle pre_{v,a}\rangle_{v\in V, a\in A})$ such that:*

- *X is a finite set of real variables.*
- *V is a finite set of locations or control modes.*
- *$init : V \to R(X)$ is the function that maps every location to an initial set.*
- *A is a finite set of actions.*
- *$inv : V \to R(X)$ defines invariants for the variables in each location.*
- *$flow : V \times \mathbb{R}^n \to R(\dot{X})$ is a flow evolution condition.*
- *$prob : V \times A \to \mathcal{P}_{fin}(\mathrm{Dist}(V \times R(X) \times \mathcal{P}(X)))$ encodes probabilistic transitions.*
- *$pre_{v,a} : prob(v, a) \to R(X)$ defines preconditions for distributions.*

For simplification of notation, we drop the subscripts of *pre* when they are clear from the context. The current state of a PHA is expressed as the current location and the values of every variables of X. Hence, a state is a pair $(v, \mathsf{a}) \in V \times \mathbb{R}^n$. If $\mathsf{a} \in inv(v)$, then we say that the state (v, a) is *admissible*. The continuous evolution is encoded in every $flow(v, \mathsf{a})$, which is a rectangle containing all valuations that the first derivative of the variables of X can take. Given $a \in A$, a transition labelled a can be taken from a state (v, a) if there is some $\mu \in prob(v, a)$ such that $\mathsf{a} \in pre_{v,a}(\mu)$, i.e., the associated precondition is satisfied. Then, the value $\mu(v', \mathsf{post}, Y)$ is the probability that given action a from location v, the system changes location to v', and the valuation of the variables is in the rectangle $\mathsf{post} \subseteq \mathbb{R}^n$. More precisely, only variables that are in Y may change value in this transition. The formal semantics will be given in Definition 2.26.

The flow function of Definition 2.24 is quite general, as it allows, for any location, to specify a different target rectangle for *any* single valuation. Most of the time, the flow is independent of the valuations and variables are often independent of each other; hence the flow function can be more simply defined from V to $R(\dot{X})$, as in the following example.

Example 2.25. *Figure 2 shows a graphical representation of the monitor of Example 2.22. The variable x represents the performance of the machine; since there is only one variable, a valuation is just a real value. The automaton is composed of three locations, Stable, Unstable, Low. The initial state is $(Stable, 0.9)$;*

hence $init(Stable) = \{0.9\}$ *and init is* $\emptyset$ *elsewhere. For the action* change *from Stable, there is one distribution* μ *satisfying* $\mu(Unstable, \{0.6\}, \{x\}) = 0.7$ *and* $\mu(Low, \{0.4\}, \{x\}) = 0.3$*; its preconditions are* $pre_{Stable,change}(\mu) = [0.85, 1]$. *The admissible states of the location Stable are of the form* $(Stable, \mathsf{a})$ *where* $\mathsf{a}(x) \in inv(Stable) = [0.8, 1]$. *For the evolution flow on Stable, the parameter* x *of admissible states evolve at the rate* $\frac{1}{1000}$*; so* $flow(Stable) = \{\frac{1}{1000}\}$. *However, in the location Unstable, the evolution is non-deterministic because* x *can evolve at any rate in the rectangle* $[-\frac{1}{64}, \frac{1}{64}]$*, so that we have* $flow(Unstable) = [-\frac{1}{64}, \frac{1}{64}]$.

Model-Checking Hybrid Systems. Because of their continuous nature, model-checking hybrid systems can only be done for sub-classes of hybrid systems. For instance, in the non-probabilistic case, Henzinger and al. [10] proposed two algorithms that allow the verification of safety properties on initialised hybrid systems. In the probabilistic case, Sproston proposed methods to verifiy some sub-classes of PHAs [14], and to verify $\forall$-PBTL on rectangular PHAs [15]. More recently, symbolic analysis have been established [9]and other approximation techniques [16].

Hybrid to Concurrent Systems. When defining PHAs, Sproston proposed a semantics in terms of probabilistic concurrent systems [14]. In these systems, time delay actions encode the continuous time nature of PHAs. They are essentially LMPs with non deterministic and discrete transition functions and with uncountably many actions. It is worth mentioning that the concurrent systems arising from a PHA have essentially a theoretical purpose: PHAs are a language to model dynamical continuous phenomena, and we need to formalise what the syntax of this language means; the semantics as concurrent systems is also used to define relations between PHAs, such as the well-known equivalence notion of bisimulation. The obtained concurrent systems cannot be used in practice, they cannot be, for example, model-checked, as they are way too general: for example, all the state changes that occur through time are represented as uncountable, unrelated non deterministic transitions, where the linearity of time is discarded.

Before defining the semantics, we give some notation[2] and insights that will help to understand the semantics of transitions. Let Q be the set of admissible states of a PHA. For each $(v, \mathsf{a}) \in Q$ and each action $a \in A$, we want to define a (possibly uncountable) set of discrete distributions in $\mathrm{Dist}(Q) \subseteq \mathrm{Dist}(V \times \mathbb{R}^n)$, from the set $prob(v, a)$, which contains distributions in $\mathrm{Dist}(V \times \mathcal{P}(\mathbb{R}^n) \times X)$. Each $\mu \in prob(v, a)$ will give rise to as many distributions on $\mathrm{Dist}(Q)$ as there are combinations $\langle \mathsf{b} \rangle := \langle \mathsf{b}_1, \mathsf{b}_2, \ldots, \mathsf{b}_m \rangle$, with $\mathsf{b}_i \in \mathsf{post}_i$ and $\mathrm{supp}(\mu) = \{(v_i, \mathsf{post}_i, Y_i)\}_{i=1}^m$. Let $Target(\mu)$ be the set of all these combinations. This is necessary, as the sets post_i may overlap and may share the same target location v_i. Every combination represents a non deterministic choice defined from μ.

Definition 2.26. *Given a PHA* $H = (X, V, init, A, inv, flow, prob, \langle pre_v \rangle_{v \in V})$*, we derive the associated infinite concurrent probabilistic system*

[2] We change slightly the notation of [14], in order to clarify the definition.

$(Q, I, A \cup \mathbb{R}, \mathit{Steps})$ as follows :

- $Q \subseteq V \times \mathbb{R}^n$ *is the set of admissible states;*
- $I = \{(v_0, \mathsf{a}_0) \in Q \mid \mathsf{a}_0 \in \mathit{init}(v_0)\}$ *;*
- $\mathit{Steps}(v, \mathsf{a}) := \mathit{Cts}(v, \mathsf{a}) \cup \mathit{Dis}(v, \mathsf{a})$*, for state* $(v, \mathsf{a}) \in Q$*, and:*
  - $\mathit{Cts}(v, \mathsf{a}) \subseteq \mathbb{R} \times Q$ *contains delay transitions: all pairs* $(d, (v, \mathsf{b}))$ *such that* $d \in \mathbb{R}_{\geq 0}$*,* $\mathsf{b} \in \mathit{inv}(v)$*, and there exists a differentiable function* $f : [0, d] \to \mathbb{R}^n$ *with* $\dot{f} : (0, d) \to \mathbb{R}^n$ *such that* $f(0) = \mathsf{a}$*,* $f(d) = \mathsf{b}$*, and for all* $\epsilon \in (0, d)$*,* $\dot{f}(\epsilon) \in \mathit{flow}(v, f(\epsilon))$ *and* $f(\epsilon) \in \mathit{inv}(v)$*;*
  - $\mathit{Dis}(v, \mathsf{a}) \subseteq A \times \mathrm{Dist}(Q)$ *contains, for each* $\mu \in \mathit{prob}(v, a)$ *with* $\mathsf{a} \in \mathit{pre}(\mu)$*, for each* $\langle \mathsf{b} \rangle \in \mathit{Target}(\mu)$*, all pairs* $(a, \mu_{\langle \mathsf{b} \rangle})$*, where* $\mu_{\langle \mathsf{b} \rangle}$ *is defined as:*

$$\mu_{\langle \mathsf{b} \rangle}(v', \mathsf{c}) := \begin{cases} \sum\limits_{\substack{i=1 \\ \mathsf{c}=\mathsf{b}_i, v_i=v'}}^{|\mathrm{supp}(\mu)|} \mu(v_i, \mathsf{post}_i, Y_i) & \text{if } \mathsf{c} \in \mathit{inv}(v') \\ 0 & \text{otherwise.} \end{cases}$$

The semantics gives us a way of using known notions of equivalence between systems such as bisimulation, simulation.

3 Aircraft Modeling

As we have seen, LMPs and PHAs both model continuous state space systems, but in a different way. The purpose of this section is to highlight those differences and the limitations of each of them. To do so, we will attempt to model the aircraft system described in the introduction with the two formalisms.

3.1 The Aircraft as an LMP

We first define a model of the aircraft benchmark using an LMP. As the aircraft system is time-continuous, this modelling cannot be done faithfully. We will have to discretize time. Let t be a unit of time, it will be our basic time delay for transitions. The set of states will of course be the possible altitudes of the aircraft together with the crash state: $[0, H_{\max}] \cup \{\mathit{Crashed}\}$, with initial state 0. The σ-algebra is the one generated by the union of $\mathcal{B}$ and the extra state: we denote it by $\mathcal{B} + \{\mathit{Crashed}\}$. The labelling function is not relevant here: it could either label all non-crashed states as *Air*, or those that are between $H_{\min}$ and $H_{\max}$ as *Safe* and the smaller ones as *Low*: this choice depends on the properties one needs to check. The set of actions will contain *rotate*, but also an action τ that will represent internal dynamics that happen at each time unit t. We define the LMP over $\mathit{Act} := \{\mathit{rotate}, \tau\}$ as:

$$G^t_{\mathrm{LMP}} = ([0, H_{\max}] \cup \{\mathit{Crashed}\}, \mathcal{B} + \{\mathit{Crashed}\}, 0, \{\mu_{\mathit{rotate}}, \mu_\tau\}, \mathit{Label}).$$

Two kinds of discrete probabilistic transitions are defined. The *rotate* transitions are defined exactly as in the specification.

– If $0 \leq s < H_{\min}$, $\mu_{rotate}(s, \{Crashed\}) = 1$.
– If $H_{\min} \leq s \leq H_{\max}$, $\mu_{rotate}(s, \cdot)$ is the unique probability measure extension of the following:

$$\mu_{rotate}(s, [a, b]) := \begin{cases} \int_a^b e^{-(s-x)} dx & \text{if } 0 < a \leq b < s \quad \text{i.e. } s - x \sim \text{Exp}(1) \\ 0 & \text{if } s \leq a \leq b \end{cases}$$

$$\mu_{rotate}(s, \{Crashed\}) := \int_{-\infty}^{0} e^{-(s-x)} dx = e^{-s}.$$

Timed transitions happen when the aircraft is rising up to its maximum altitude. The specification says that this is done with a rate between 20 and 25. Hence, if the altitude is $0 \leq s \leq H_{\max} - 25$, the aircraft should end up at an altitude between $s + 20t$ and $s + 25t$ after the time unit t has elapsed. However, this uncertainty is unquantified, which makes it impossible to be modelized as it is in the LMP structure. Hence we have to make a choice on how it will happen. We choose the uniform distribution. Hence,

$$\text{if } 0 \leq s \leq H_{\max} - 25t, \quad \mu_\tau(s, \cdot) := U(s + 20t, s + 25t)$$

For s between $H_{\max} - 25t$ and $H_{\max}$, μ_τ is defined in the obvious way, in order to not exceed the maximal altitude $H_{\max}$. The obtained LMP G^t_{LMP} is depicted in Figure 3. An example of property that can be verified on G^t_{LMP} is **P1**:"if $s > H_{\min}$, the probability that the aircraft loses 100 meters or more when rotating is between 25% and 50%". The distribution associated to the action *rotate* if $s > H_{\min}$ gives a probability of $p(s, [0, s-100] \cup \{Crashed\}) = e^{-1} - e^{-s} + e^{-s} = e^{-1}$. Hence we conclude that G^t_{LMP} satisfies the property. However, any property that needs an accuracy greater than the one we have chosen with t will not be verified accurately. For example consider the following simple property, **P2**:"if the state is above $H_{\min}$, it takes less than a second to gain 10 meters".

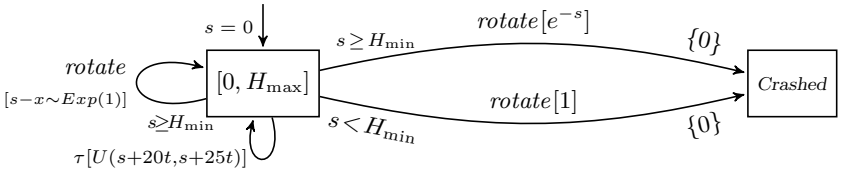

Fig. 3. The aircraft as an LMP

3.2 The Aircraft as a Rectangular PHA

We now want to model the aircraft by way of a PHA. As the system is a time-continuous system, hybrid systems are perfect for modeling delay transitions. On the other hand, PHAs allow only discrete probability distributions whereas the aircraft probability distributions are continuous. Hence, the aircraft system can not be modeled faithfully once again. In the PHA model, there will be only one variable, the altitude, hence valuations are just real values. There will be two

locations, *Air*, representing the situation where the aircraft can fly and *Crashed*. There is one action, *rotate*. We now define the remaining parameter of our PHA model, that we denote by:

$$G_{\text{PHA}} = (\{x\}, \{Air, Crashed\}, init, \{rotate\}, inv, flow, prob, \langle pre_{Air}, pre_{Crashed}\rangle)$$

- $init(Air) = \{0\}$.
- $inv(Air) = [0, H_{\max}]$ and $inv(Crashed) = \{0\}$.
- $flow(Air) = [0.2, 0.25]$ and $flow(Crashed) = \{0\}$.
- $prob(Crashed, rotate) = \emptyset$, as no action can be done.
- $prob(Air, rotate)$ has to be split in two distributions:
  - μ_1 with $pre(\mu_1) = [0, H_{\min}]$: for all $s \in [0, H_{\min}]$, $\mu_1(Crashed, \{0\}, \{s\}) = 1$, that is, the aircraft ends up in state *Crashed* and s is set to 0.
  - μ_2 with $pre(\mu_2) =]H_{\min}, H_{\max}]$: for all $s \in pre_{Air}(\mu_2)$, the definition of $\mu_2((Air, \mathsf{post}, Y))$ is discussed below.

In defining the distribution μ_2, we have a difficult choice to make. Indeed, to be faithful to the model, we should specify a precondition for every value of x and we would need μ_2 to be non discrete. This is not possible, as we are only allowed a *finite* number of *discrete* distributions. One solution is to partition the interval $]H_{\min}, H_{\max}]$ into as many intervals $I_1, I_2, \ldots, I_n$ as possible, split the *Air* location with respect to these intervals and then, for the transition from I_j to I_k, written $\mu_j(Air, I_k, \emptyset)$, choose a value that would represent the set $\{p(i, I_k) \mid i \in I_j\}$ (where p is the probability defined for the aircraft in the introduction). To simplify, we choose to not split the location for preconditions, but we do consider apart the two post conditions $I_1 := [0, H_{\min}]$ and $I_2 :=]H_{\min}, H_{\max}]$; we must choose values that represent the sets $P_1 := \{p(x, [0, H_{\min}]) \mid x \in [H_{\min}, H_{\max}]\}$ and $P_2 := \{p(x,]H_{\min}, H_{\max}]) \mid x \in [H_{\min}, H_{\max}]\}$. We assign parameter values here of q and q' to simplify, while not making any assumption (one could choose the infima or the mean of the sets P_1 and P_2). In the same way, we assign value q'' to $\mu_2(Crashed, \{0\}, \{x\})$. Figure 4 shows the obtained PHA.

Of course, we are quite far from the exponential distribution that was required in the specification of the aircraft. We now discuss how well this model

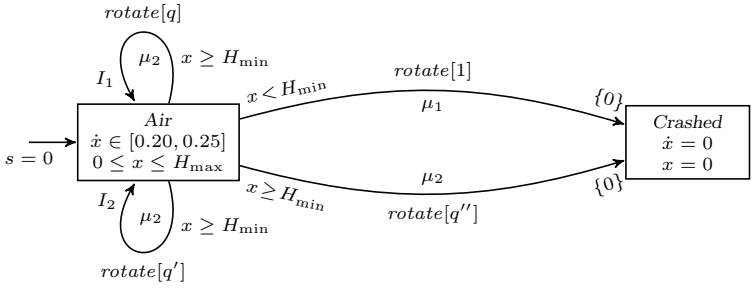

Fig. 4. The aircraft as a PHA: here $I_1 = [0, H_{\min}]$ and $I_2 =]H_{\min}, H_{\max}]$

can be used to check our properties **P1** and **P2**. Recall **P1**:"if $s > H_{\min}$, the probability that the aircraft loses 100 meters or more when rotating is between 25% and 50%". This property could be checked in the LMP version of the aircraft. However, in this model, all we can say about the probability to reach $[0, s-100] \cup \{Crashed\}$ is that it is greater than the probability to reach $[0, H_{\min}] \cup \{Crashed\} = q' + q''$. Thus we will be able to refute the formula if $q' + q''$ is greater than 50%, but otherwise we cannot say anything. On the other hand, property **P2**: "if the state is above $H_{\min}$, it takes less than a second to gain 10 meters" is verified on G_{PHA}.

3.3 Conclusion

We have seen in this case study how to approximate the aircraft example with LMPs and PHAs. We can conclude that neither can faithfully model systems such as the aircraft. There are two main divergences between the two models; they lie in the nature of distributions and in the time class they belong to.

It is important to notice that the failure of PHAs to model faithfully the exponential distribution on our case study is not due to the limitation of rectangular PHAs. Even with general PHA, the *shape* of the definition is inadequate to model continuous distributions. This leads us to highlight a very important observation about LMPs: the continuity of the state space in LMPs crucially depends on the continuous nature of the distribution. Restricting to discrete distribution on any space will result in a discrete process (up to bisimulation).

In the light of those divergences, we propose, in the next section, a generalization that reflects the characteristics of both LMPs and PHAs.

4 Hybrid Labelled Markov Processes

4.1 Definition

Hybrid Labelled Markov Processes (HLMP) combine both Labelled Markov Processes and probabilistic hybrid systems behaviors.

Our starting point is PHAs. We also choose to keep the set of locations finite instead of taking an arbitrary measure space (S, Σ). This is a theoretical restriction that should not prevent further applications and that helps to keep the model tractable. The crucial modification will be in how we will integrate continuous distributions. To do so, it is not sufficient to just permit $prob(v, a)$ to be a finite set of continuous distributions. At first sight, by doing this, the behavior of v would indeed become continuous. However, because this set would only depend on v and the preconditions that we would further attach to this distribution, we would force continuously many valuations, and hence continuously many states, to behave as v does. By doing so, all these states could end up being bisimilar in the underlying semantics and hence we would be back at our starting point: a discrete distribution. In particular, observe that the exponential distribution in the aircraft example could not be modelled, since it depends on

the current value of the altitude and not on the fact that the aircraft is either flying or has crashed.

Consequently, there are two features of a PHA that have to be modified in order to insert continuous distributions to the model: the *prob* and *pre* functions. What is encoded in the precondition function in the PHA will become a parameter of the probabilistic transition function. Hence the subprobability measures that will be returned by *prob* will have the possibility to use this parameter. We also need to define a non standard set of subprobability measures: given a finite set V and a measurable space (S, Σ),

$$\overline{\mathsf{Sub}}(S, V \times \Sigma) := \{\mu \in \mathsf{Sub}(S, V \times \Lambda) \mid \Lambda \text{ is a } \sigma\text{-algebra and } \Lambda \subseteq \Sigma\}.$$

We define an Hybrid Labelled Markov Process as follow:

Definition 4.11. *An Hybrid Labelled Markov Process (HLMP) is a structure* $M = (X, V, init, A, inv, flow, prob, Label)$ *defined as follows:*

- X, V, A *are as in Definition 2.24*
- $init : V \to \mathcal{P}(\mathbb{R}^n)$ *defines an initial set*[3]*.*
- $inv : V \to \mathcal{P}(\mathbb{R}^n)$ *defines invariants for the variables in each location.*
- $flow : V \times \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^n)$ *is a flow evolution condition.*
- $prob : V \times \mathbb{R}^n \times A \to \mathcal{P}_{fin}(\overline{\mathsf{Sub}}(V \times \mathbb{R}^n, V \times \mathcal{B}_n))$ *encodes probabilistic transitions.*
- $Label : V \times \mathbb{R}^n \to \mathcal{P}(AP)$ *is a measurable function used to describe states.*

Let us explain the *prob* function. First observe that in PHAs, there is non determinism coming from a distribution μ if there is a set post of more than one valuations such that $\mu(v, \mathsf{post}, Y) > 0$. It is because we do not have specific probabilities for subsets of post that non deterministic transitions arise. Aside from this, sets of this form can overlap and we have to take account of the multiple copies produced when defining the semantics. This can be done quite easily because the distributions are discrete. With continuous distributions, if post_i are measurable sets, $i = 1, 2$, then the probability of both of them, as well as their union and intersection will be known. Hence, if for some state s and action a the σ-algebra Σ on which $\mu \in prob(s, a)$ is defined is $\mathcal{P}(V) \times \mathcal{B}_n$, then there is no non determinism in μ: every measurable set of this σ-algebra can be split into smaller sets (unless it is a singleton) whose probability will be determined by μ. On the other hand, when μ is defined on a smaller σ-algebra $\Lambda \subseteq \mathcal{P}(V) \times \mathcal{B}_n$, there may be *minimal* sets E (called *atoms* of the σ-algebra), for which $\mu(E)$ is defined, but no set included in E gets a value from μ: this corresponds to the same non determinism that happens in the sets "post" of PHAs.

Depending on the differential equation characterizing the flow evolution, we can distinguish the same kind of sub-classes as for PHAs. For instance, if the parameters $x \in X$ evolve with respect to equations $\dot{x} = k$ for $k \in \mathbb{Q}$, then the system would be a *linear HLMP*. Moreover, a rectangular HLMP is obtained if the initialization, the invariants, the flow evolution, the preconditions and the postconditions sets are subsets of $R(X)$.

[3] We write it this way for simplicity, but the image is rather $\mathcal{B}_n$.

He now define a semantics for HLMPs, and for this we must redefine the *Target* function. Let $\mu \in prob(v, \mathsf{a}, a)$ be defined on the σ-algebra $V \times \Lambda$ and let $\{\mathsf{E}_i\}_{i \in I}$ be the family of atoms of Λ such that $\mu(v, \mathsf{E}_i) > 0$ for some $v \in V$. This family can be at most countable (because $\mu(V \times \mathbb{R}^n) \leq 1$). We define $Target(\mu)$ as the set of all combinations $\langle \mathsf{b} \rangle := \langle \mathsf{b}_1, \mathsf{b}_2, \ldots, \mathsf{b}_m \rangle$, with $\mathsf{b}_i \in \mathsf{E}_i$. We also need to augment the σ-algebra Λ with these valuations which are not, as singletons, part of it; hence, we define $\Lambda^+ := \Lambda \cup \sigma(\{\{\mathsf{b}\} \mid \exists i \in I. \mathsf{b} \in \mathsf{E}_i\})$[4].

Definition 4.12. *Given a HLMP $H = (X, V, init, A, inv, flow, prob)$, we derive the associated infinite concurrent probabilistic system*

$$(Q, Init, A \cup \mathbb{R}, Steps) \text{ as follows :}$$

- $Q := \{(v, \mathsf{a}) \in V \times \mathbb{R}^n \mid \mathsf{a} \in inv(v)\}$ *is the set of admissible states;*
- $Init = \{(v_0, \mathsf{a}_0) \in Q \mid \mathsf{a}_0 \in init(v_0)\}$ *;*
- $Steps(v, \mathsf{a}) := Cts(v, \mathsf{a}) \cup Dis(v, \mathsf{a})$*, for state $(v, \mathsf{a}) \in Q$, and:*
  - *Cts is defined as in Definition 2.26;*
  - *$Dis(v, \mathsf{a}) \subseteq A \times \mathsf{Sub}(Q)$ contains for each $\mu \in prob(v, \mathsf{a}, a)$ and $\langle \mathsf{b} \rangle \in Target(\mu)$, all pairs $(a, \mu_{\langle \mathsf{b} \rangle})$ such that $\mu_{\langle \mathsf{b} \rangle}$ is defined as, for $\mathsf{C} \in \Lambda^+$:*

$$\mu_{\langle \mathsf{b} \rangle}(v', \mathsf{C}) := \mu(v', \mathsf{C} \setminus \cup_i \mathsf{E}_i) + \sum_{\substack{i \in I \\ \mathsf{b}_i \in \mathsf{C} \cap inv(v')}} \mu(v', \mathsf{E}_i).$$

It is easy to see that HLMPs encompass PHAs. It is only because of the choice of taking the set of locations finite that they do not include LMPs as well. However, they share with LMPs the same intrinsic continuity with respect to distributions.

4.2 Aircraft Modeling

We now show that this new framework of HLMP permits to model exactly the aircraft example of Section 1.1. The wanted model is

$$G_{\text{HLMP}} = (\{x\}, \{Air, Crashed\}, init, \{rotate\}, inv, flow, prob)$$

- X, V , *init*, A, *inv*, *flow* are the same as G_{PHA}.
- *prob* is defined as follows
  - if $x \leq H_{\min}$, $prob(Air, x, rotate)$ contains only the distribution that gives probability 1 to *Crashed* and 0 elsewhere.
  - if $H_{\min} \leq x \leq H_{\max}$, $prob(Air, x, rotate) = \{\mu_x\}$ where $\mu_x(Crashed, 0) = e^{-x}$ and $\mu_x(Air, E) = \mu_{rotate}(E)$ if $E \subseteq \mathbb{R}$, where μ_{rotate} is from the LMP modelisation of the aircraft (Section 3.1).

This representation combines both the representations as a PHA of Section 3.2 and as the LMP of Section 3.1. Figure 5 shows a graphical representation of this model of the aircraft. In this model, both properties **P1** and **P2** can be verified accurately. Moreover, the following property can also be checked, whereas it could not be checked accurately by neither the LMPs nor the PHAs models: **P3** "if a rotation happens within 20 to 30 seconds, there will be no crashing with probability greater than 1/4". Intuitively, the computation results in the non-empty set $\{(Air, x) \mid x \in [5, 7.5]\}$ of states reachable from the initial state, and therefore **P3** is satisfied by G_{HLMP}.

[4] If W is a set of sets, $\sigma(W)$ is the smallest σ-algebra containing W.

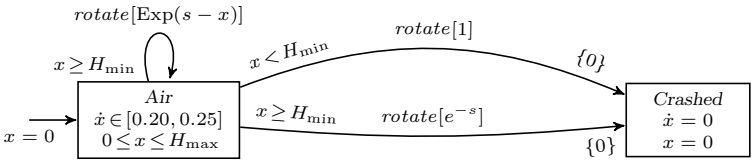

Fig. 5. The HLMP model of the aircraft

4.3 Model-Checking of HLMP

We discuss how the verification of HLMP can be done. We propose two approaches. Recall that only some sub-classes of hybrid systems (probabilistic or not) can be verified. The simplest approach we propose to verify a HLMP is under implementation and is to construct an LMP from the HLMP by discretizing the time in the same fashion as we did to model the aircraft using LMPs, in Section 3.1. If we have some reasonable unit of time, or if we know in advance the properties to be checked, we can deduce a unit of time t that will make the time discretization a reasonable approximation of the given HLMP with respect to the properties. More precisely, for each state s, only one delay transition will be enabled, a τ-transition of delay t, in the same way as for the aircraft. The result is a narrower process than the one obtained through the semantics of HLMPs where one d-transition for every value $d \in \mathbb{R}$ is defined. The probabilistic part of transitions, needs no transformation. Yet there is one more thing to be precised here, it is that the verification will be faithful to the HLMP if this one contains no non determinism. This requires on one hand that the HLMP be linear, that is, if all flow equations are of the form $\dot{x} = c$ for some real value c: this way, the state reached after the delay t is determined. On the other hand, to not contain non determinism also requires that probabilistic transitions be defined on the total σ-algebra of states. Otherwise, the non determinism will have to be resolved in the construction of the LMP. It will depend on the context to decide if time-discretization is adequate by providing a suitable distribution over the rectangles. Finally, with an LMP in hand, we can use the methodology defined for them. This include finite approximations [3], distances [4,8], and a model checker [13]. One could wonder why defining HLMPs if we are to further translate them into LMPs. We believe that having a formalism to specify a system exactly is by itself important. Whether there are techniques to verify the models either exactly, approximately, automatically or semi-automatically is another issue. One reason is that an inexact model of the system can be suitable at some point but not later on: with only the approximation at hand, we must start over to build a new one, possibly finer, for another use. Actually, we want this process to be transparent to the user: an automated tool can construct an LMP from an HLMP – and a time unit – in polynomial time. Alternatively, given an HLMP and a property to check, the same tool could choose an appropriate time unit.

The other approach we propose is more involved and needs some more work. HLMPs could be checked by adapting known algorithms for PHAs and LMPs to

them. In the probabilistic case, methods to verifiy some sub-classes of PHAs have been proposed [14,15]. More recently, symbolic analysis have been established [9]. These methods, combined with the methodology and tool of Richard [13] should extend to HLMPs. We leave this for future work but we believe that constructing the underlying mathematical framework needed for this will probably consist in combining the existing techniques for PHAs and LMPs.

5 Conclusion

The main purpose of this paper was to compare two models involving continuous state spaces, the continuity arising from different features in each framework. We observed that LMPs' is inherent to the probability distributions of transitions, whereas PHAs' continuity is in the evolution of the model's state. The two models have evolved independently and this paper shows on one hand that they are incomparable, and on the other hand that they are nevertheless compatible, and hence there is a need for a unifying framework. We compared these models through a new case study that permits to highlight their differences and limitations. The aircraft example, despite its simplicity – as only one action and only its altitude are observed – is a continuous process that cannot be modelled faithfully by neither LMPs nor PHAs. Some approximations can be defined, and we exhibited one for each. For both approximations, we also expressed properties that can be verified in one approximation but not in the other.

From the observation that none of the formalisms considered could faithfully model the aircraft, there was only one step to propose a generalization of both to circumvent those limitations. Hence, we defined hybrid LMPs and their semantics. This formalism can model systems that combine both continuous time and continuous distributions. Of course, it can model our aircraft case study. Finally, we proposed verification approaches that use the existing verification techniques of both formalisms. Of course, a lot of work is to be done to extend the theory of LMPs and PHAs to the world of HLMPs.

Acknowledgement

J. Desharnais wishes to thank Marta Kwiatkowska and Oxford University for welcoming her during the current year.

References

1. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Journal Theoretical Computer Science 138 (1995)
2. Blute, R., Desharnais, J., Edalat, A., Panangaden, P.: Bisimulation for labelled Markov processes. In: Proc. of the Twelfth IEEE Symposium on Logic in Computer Science, Warsaw, Poland (1997)

3. Bouchard-Cote, A., Ferns, N., Panangaden, P., Precup, D.: An approximation algorithm for labelled Markov processes: towards realistic approximation. In: QEST 2005: Proc. of the Second International Conference on the Quantitative Evaluation of Systems, p. 54. IEEE Computer Society, Washington (2005)
4. van Breugel, F., Sharma, B., Worrell, J.: Approximating a Behavioural Pseudometric Without Discount for Probabilistic Systems. In: Seidl, H. (ed.) FOSSACS 2007. LNCS, vol. 4423, pp. 123–137. Springer, Heidelberg (2007)
5. Cattani, S., Segala, R., Kwiatkowska, M., Norman, G.: Stochastic transition systems for continuous state spaces and non-determinism. In: Sassone, V. (ed.) FOSSACS 2005. LNCS, vol. 3441, pp. 125–139. Springer, Heidelberg (2005)
6. D'Argenio, P.R., Wolovick, N., Terraf, P.S., Celayes, P.: Nondeterministic Labeled Markov Processes: Bisimulations and Logical Characterization. In: International Conference on Quantitative Evaluation of Systems, pp. 11–20 (2009)
7. Desharnais, J., Gupta, V., Jagadeesan, R., Panangaden, P.: Approximating Continuous Markov Processes. In: Proc. of the 15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara, Californie, USA, pp. 95–106 (2000)
8. Desharnais, J., Laviolette, F., Zhioua, S.: Testing Probabilistic Equivalence through Reinforcement Learning. In: Proc. of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science, pp. 664–677 (2006)
9. Fränzle, M., Hermanns, H., Teige, T.: Stochastic Satisfiability Modulo Theory: A Novel Technique for the Analysis of Probabilistic Hybrid Systems. In: Egerstedt, M., Mishra, B. (eds.) HSCC 2008. LNCS, vol. 4981, pp. 172–186. Springer, Heidelberg (2008)
10. Henzinger, T.A., Ho, P.-H., Wong-toi, H.: Algorithmic Analysis of Nonlinear Hybrid Systems. Journal IEEE Trans. on Automatic Control 43, 225–238 (1996)
11. Hassapis, G., Kotini, I.: Verification of rectangular hybrid automata models. Journal The Journal of Systems and Software 79(10) (2006)
12. Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: An Approach to the Description and Analysis of Hybrid Systems. In: Grossman, R.L., Ravn, A.P., Rischel, H., Nerode, A. (eds.) HS 1991 and HS 1992. LNCS, vol. 736, pp. 149–178. Springer, Heidelberg (1993)
13. Richard, N.: Labelled Markov Processes. Master thesis, Département d'informatique et de génie logiciel, Université Laval (2003)
14. Sproston, J.: Analyzing Subclasses of Probabilistic Hybrid Automata. In: Proc. of the 2nd International Workshop on Probabilistic Methods in Verification, Eindhoven. University of Birmingham, Technical Report, CS-99-8 (1999)
15. Sproston, J.: Model Checking of Probabilistic Timed and Hybrid Systems. Ph.D. thesis, University of Birmingham, Faculty of Science (2000)
16. Zhang, L., She, Z., Ratschan, S., Hermanns, H., Hahn, E.M.: Safety verification for probabilistic hybrid systems. In: Touili, T., Cook, B., Jackson, P. (eds.) Computer Aided Verification. LNCS, vol. 6174, pp. 196–211. Springer, Heidelberg (2010)