Measurability and Safety Verification for Stochastic Hybrid Systems

Martin Fränzle
Carl von Ossietzky University, Germany

Ernst Moritz Hahn
Saarland University, Germany

Holger Hermanns
Saarland University, Germany

Nicolás Wolovick
National University of Córdoba, Argentina

Lijun Zhang
Technical University of Denmark

ABSTRACT

Dealing with the interplay of randomness and continuous time is important for the formal verification of many real systems. Considering both facets is especially important for wireless sensor networks, distributed control applications, and many other systems of growing importance. An important traditional design and verification goal for such systems is to ensure that unsafe states can never be reached. In the stochastic setting, this translates to the question whether the probability to reach unsafe states remains tolerable. In this paper, we consider stochastic hybrid systems where the continuous-time behaviour is given by differential equations, as for usual hybrid systems, but the targets of discrete jumps are chosen by probability distributions. These distributions may be general measures on state sets. Also non-determinism is supported, and the latter is exploited in an abstraction and evaluation method that establishes safe upper bounds on reachability probabilities. To arrive there requires us to solve semantic intricacies as well as practical problems. In particular, we show that measurability of a complete system follows from the measurability of its constituent parts. On the practical side, we enhance tool support to work effectively on such general models. Experimental evidence is provided demonstrating the applicability of our approach on three case studies, tackled using a prototypical implementation.

Categories and Subject Descriptors: I.6.4 [Computing Methodologies]: Simulation and Modelling - Model Validation and Analysis; C.1.m [Computer Systems Organization]: Processor Architectures - Hybrid Systems; G.3 [Mathematics of Computing]: Probability and Statistics
General Terms: Reliability, Verification.

1. INTRODUCTION

In many modern application areas of hybrid systems, *random phenomena* occur. This is especially true for wireless sensing and control applications, where message loss probabilities and other random effects (node placement, node failure, battery drain, measurement imprecision) turn the overall control problem into a problem that can only be managed with a certain, hopefully sufficiently large, probability. The need to integrate probabilities into hybrid systems formalisms has led to several different notions of *stochastic hybrid systems*, each from a distinct perspective [2, 3, 9, 22]. They differ in the point of attack where to introduce randomness. One option is to replace deterministic jumps by probability distributions over deterministic jumps. Another option is to generalise the differential equation components inside a mode by a stochastic differential equation component. More general models can be obtained by blending the above two choices, and by combining them with memoryless timed probabilistic jumps [8], and with non-determinism. Piecewise-deterministic Markov processes [12] are a prominent example, constituting deterministic hybrid system models augmented with memoryless timed probabilistic jumps.

An important problem in hybrid systems theory is that of reachability analysis. In general terms, a reachability analysis problem consists in evaluating whether a given system will reach certain unsafe states, starting from certain initial states. This problem is associated with the safety verification problem: if the system cannot reach any unsafe state, then the system is declared to be safe. In the probabilistic setting, the safety verification problem can be formulated as that of checking whether the probability that the system trajectories reach an unsafe state from its initial states can be bounded by some given probability threshold.

Recently [23], we have introduced a technique that piggybacks a quantitative probabilistic reachability analysis for *probabilistic* hybrid automata on a qualitative reachability checker for *non-probabilistic* hybrid automata. It does so to compute upper bounds on maximal reachability probabilities, but is restricted to probability distributions with finite support. In this paper, we extend the approach to stochastic hybrid automata which feature *continuous* measures over states, for instance given by a density function, as well as non-deterministic behaviour. We show that the well-definedness of the individual automata parts leads to the well-definedness of the model semantics, such that we can define meaningful probabilities when resolving the non-determinism. For this we harvest recent results on non-deterministic labelled Markov processes [11] to overcome intricate measurability issues. To handle reachability problems for this model class, we over-approximate the stochastic hybrid automaton by a probabilistic hybrid automaton, in which the probability to reach unsafe states can not be lower

HSCC'11, April 12–14, 2011, Chicago, Illinois, USA.

than in the original model. Because this transformation is done on the high-level description of the automata, we can combine it with our previous bounding technique and algorithmic implementation. We are thus able to tackle a broad class of stochastic hybrid systems. Due to the presence of continuous non-determinism, but absence of diffusion in the differential equation components, the model is distinguished from more classical stochastic hybrid systems representations for which reachability computations have been tackled in different ways [1, 2, 9, 19, 20]. Among them, grid-based methods [1] are a promising approach with some similarities to our work. In contrast to those, we do not compute a fixed state-space abstraction prior to the actual analysis. Instead, an abstraction is computed by a solver for reachability in a non-probabilistic version of the hybrid systems under consideration. From the abstraction obtained this way, we compute a probabilistic model which faithfully over-approximates the actual maximal reachability probability.

The paper is organised as follows: Section 2 describes the model which will later on form the semantics of stochastic hybrid automata. Then, in Section 3, we give the high-level specification model of stochastic hybrid automata. Afterwards, in Section 4 we describe how to compute over-approximating reachability probabilities in an abstracted model of the automata. The practical applicability of the method is demonstrated in Section 5. Finally, we conclude the paper in Section 6.

2. SEMANTIC MODELS

We define non-deterministic Markov processes, which shall later appear as the semantics of stochastic hybrid automata. Subsequently, we show how to abstract these models, which potentially feature continuous measures, to probabilistic automata with measures of only finite support. For the specification of these models, we need some preliminary definitions of measure theoretic concepts.

Measure Theory Background.

A family Σ of subsets of the set S is a σ-algebra provided it is closed under complement and σ-union (denumerable union), a set $A \in \Sigma$ is then called measurable. We denote by $\sigma(\mathcal{A})$ the smallest σ-algebra containing the sets of $\mathcal{A}$, and it is said to be generated by this set. The Borel σ-algebra over $\mathbb{R}$ is generated by intervals of rational endpoints, and it is denoted $\mathcal{B}(\mathbb{R}) := \sigma(\{[p,q) \mid p,q \in \mathbb{Q}\})$. For dimensions greater than one, it is generated by rectangles with rational endpoints and denoted by $\mathcal{B}(\mathbb{R}^n)$. The pair (S,Σ) is called a measurable space, and where convenient we will use $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$ to denote the particular Borel case.

Given two measurable spaces (S_0,Σ_0), (S_1,Σ_1), the product space is given by $(S_0 \times S_1, \Sigma_0 \otimes \Sigma_1)$, where the product σ-algebra is generated by rectangles $A_0 \times A_1$, with $A_i \in \Sigma_i$. Given $M \in \Sigma_0 \otimes \Sigma_1$, the section at s_0 is the measurable set $M_{|s_0} := \{s_1 \mid (s_0,s_1) \in M\}$.

A function $\mu : \Sigma \to [0,1]$ is called σ-additive if $\mu(\biguplus_{i\in I} A_i) = \sum_{i\in I} \mu(A_i)$ for countable index sets I. We speak of a probability measure if $\mu(S) = 1$. The support of a measure $Supp(\mu)$ is a measurable set A such that $\mu(S \setminus A) = 0$, and for the Borel σ-algebra there is a (unique) smallest closed set C^0 with this property. The Dirac probability measure δ_s is 1 only in $\{s\}$. A function is measurable, denoted $f : (\mathbb{R},\mathcal{B}(\mathbb{R})) \to (\mathbb{R},\mathcal{B}(\mathbb{R}))$ in the real-valued case, if every backwards image of a generator is measurable, $f^{-1}([p,q)) \in \mathcal{B}(\mathbb{R})$ for all rational p and q. The indicator function $\chi_A : S \to \{0,1\}$ is $\chi_A(x) := 1$ iff $x \in A$, and it is measurable if $A \in \Sigma$. A function f is *simple* if it is of the form $f(x) = \sum_{i=1}^n c_i \chi_{A_i}(x)$, and if $c_i \in \mathbb{R}$ and $A_i \in \Sigma$ then f is also measurable. Without loss of generality, we can assume pairwise disjoint A_i. The set of probability measures $\Delta(S)$ on (S,Σ) can be endowed with σ-algebra $\Delta(\Sigma)$ [14] generated by the set $\Delta^{>q}(A) := \{\mu \mid \mu(A) > q\}$, i.e. the measures such that when applied to $A \in \Sigma$ give a value greater than $q \in \mathbb{Q} \cap [0,1]$. In order to define measurable functions $f : S \to \Delta(\Sigma)$, we need to define the σ-algebra on $\Delta(\Sigma)$. We use [11] $H(\Delta(\Sigma)) := \sigma(\{H_\Phi \mid \Phi \in \Delta(\Sigma)\})$, where the *hit set* is $H_\Phi := \{\xi \in \Delta(\Sigma) \mid \xi \cap \Phi \neq \emptyset\}$, such that f is measurable if $f^{-1}(H_\Phi) \in \Sigma$, and we write $f : (S,\Sigma) \to (\Delta(\Sigma), H(\Delta(\Sigma)))$.

We let $\Delta_f(S)$ denote the set of finite measures, which contains all μ such that $|Supp(\mu)| < \infty$. For a singleton set $\{a\} \in Supp(\mu)$ we write $a \in Supp(\mu)$, $\mu(a) = \mu(\{a\})$, and so on.

We can now define our stochastic models.

Definition 1. *A* non-deterministic Markov process *(NMP) $\mathcal{M}$ is a tuple $(S,\Sigma,Init,Steps,UnSafe)$ where*

- *S denotes the (possibly uncountable) set of states,*
- *Σ is a σ-algebra on S. We use the subset $Init \in \Sigma$ to specify the set of* initial *states and $UnSafe \in \Sigma$ for the set of* unsafe *states.*
- *$Steps : (S,\Sigma) \to (\Delta(\Sigma), H(\Delta(\Sigma)))$ is a measurable transition function. We require that $Steps(s) \neq \emptyset$ for all $s \in S$.*

A probabilistic automaton *(PA) is an NMP where transition functions are restricted to finite support measures, $Steps : S \to 2^{\Delta_f(S)}$. For PA, measurability considerations are not needed to specify meaningful probabilities.*

This definition is essentially the same as the one of D'Argenio et al. [11], with the difference that we do not use action labels, as they are not needed in our setting. If $\mu \in Steps(s)$, we call μ a *successor probability measure* of s. Measurability considerations are indeed necessary for general NMP, as we demonstrate in the following example.

Example 1. *Let $(S,\mathcal{B}(S),\{s_0\},Steps,\{s_b\})$ be an NMP with $S := \{s_0\} \uplus [0,1] \uplus \{s_b\}$. We define $Steps(s_0) := \{\mu\}$, where μ is the uniform distribution on $[0,1]$, and let $Steps(s_b) := \{\delta_{s_b}\}$. Further, for $s \in \mathcal{V}$ we let $Steps(s) := \{\delta_{s_b}\}$ and for $s \notin \mathcal{V}$ we define $Steps(s) := \{\delta_s\}$, where $\mathcal{V} \subset [0,1]$ is a Vitali set, which is known not to be Borel-measurable. All $Steps(\cdot)$ are singleton sets, so there is no non-determinism in the example. Thus, the probability to reach s_b from s_0 should be uniquely defined. However, this probability does not exist, because it depends on the measure of the non-measurable Vitali set. Discrete measures (which are the only ones allowed in PA) eliminate the Vitali set problem, since they discard all but a finite set of points out of it. For instance, if instead of using the uniform distribution we set $\mu(s_1) = \mu(s_2) = \mu(s_3) = \frac{1}{3}$ for $s_1, s_2 \in \mathcal{V}$ and $s_3 \notin \mathcal{V}$ and $\mu(\cdot) = 0$ else, we can specify the reachability probability as $\frac{2}{3}$, although formally the model still does not fulfil the measurability requirements.*

PAs have been introduced by Segala and Lynch [21]. For PAs, we can always assign a probability larger than zero

to each individual state which is possibly chosen as successor. This is not the case for general NMPs: all individual probabilities of moving to a successor state may be zero.

For an NMP $\mathcal{M} = (S, \Sigma, \mathit{Init}, \mathit{Steps}, \mathit{UnSafe})$, we specify the maximal n-step probability to reach the unsafe states, starting in a state $s \in S$, as $\mathit{Reach}^{\mathcal{M}}_{\leq 0}(s) = 1$ if $s \in \mathit{UnSafe}$ and 0 else,

$$\mathit{Reach}^{\mathcal{M}}_{\leq n+1}(s) := \begin{cases} 1 \text{ if } s \in \mathit{UnSafe}, \\ \sup_{\mu \in \mathit{Steps}(s)} \int \mathit{Reach}^{\mathcal{M}}_{\leq n}(s') \, \mathrm{d}\mu(s') \text{ else}, \end{cases}$$

where $\int$ denotes Lebesgue integration. For a PA $\mathcal{M} = (S, \Sigma, \mathit{Init}, \mathit{Steps}, \mathit{UnSafe})$, we can simplify the latter formula to

$$\mathit{Reach}^{\mathcal{M}}_{\leq n+1}(s) = \begin{cases} 1 \text{ if } s \in \mathit{UnSafe}, \\ \sup_{\mu \in \mathit{Steps}(s)} \sum_{s' \in \mathit{Supp}(\mu)} \mathit{Reach}^{\mathcal{M}}_{\leq n}(s') \mu(s') \text{ else}. \end{cases}$$

The maximal unbounded reachability probability is

$$\mathit{Reach}^{\mathcal{M}}(s) := \lim_{n \to \infty} \mathit{Reach}^{\mathcal{M}}_{\leq n}(s).$$

These definitions indeed define functions in n and s:

LEMMA 1. *Let $\mathcal{M}$ be an NMP. Then $\mathit{Reach}^{\mathcal{M}}_{\leq n}(s)$ is well-defined for all non-negative n, as is $\mathit{Reach}^{\mathcal{M}}(s)$.*

We now specify approximations between general NMPs and PAs operating on the same state set.

DEFINITION 2. *Let $\mathcal{M}_c = (S, \Sigma, \mathit{Init}_c, \mathit{Steps}_c, \mathit{UnSafe}_c)$ be an NMP, and $\mathcal{M}_f = (S, \Sigma, \mathit{Init}_f, \mathit{Steps}_f, \mathit{UnSafe}_f)$ be a PA. We say $\mathcal{M}_f$ is an* abstraction *of $\mathcal{M}_c$, if $\mathit{Init}_c \subseteq \mathit{Init}_f$, $\mathit{UnSafe}_c \subseteq \mathit{UnSafe}_f$, and moreover, for each $s \in S$ and $\mu_c \in \mathit{Steps}_c(s)$,*

- *there exist pairwise disjoint $A_1, \ldots, A_n \in \Sigma$ such that $\mu_c(\bigcup_{i=1}^n A_i) = 1$, and*
- *in the PA, for each $(s_1, \ldots, s_n) \in A_1 \times \cdots \times A_n$ there exists $\mu_f \in \mathit{Steps}_f(s)$ such that $\mu_f(s_i) = \mu_c(A_i)$ for all $1 \leq i \leq n$.*

We write $\mathcal{M}_c \preceq_{cf} \mathcal{M}_f$ if $\mathcal{M}_f$ is an abstraction of $\mathcal{M}_c$.

Notably, and in contrast to related abstraction methods [1], we do not fix a representative of the probability measure, but instead introduce uncountable non-determinism over the possible successors. While this may seem unfamiliar and impractical, indeed it is not. The models under consideration may anyway have an uncountably large state-space, such that a direct analysis is impossible. Later on, semantics of stochastic hybrid automata will be given as uncountably large NMPs. However, we will show how to compute abstractions of the semantics directly from the high-level description of the model. Thereby, we will construct a finitely large state-space, and represent the uncountable non-determinism by a finite number of transitions.

For PAs, simulation preorders have been introduced [21], and subsequently exploited [23], to analyse safety properties. When restricted to this subclass, our notion of abstraction establishes a special case of such simulation preorders.

Applying the abstraction from Definition 2 does not decrease the maximal reachability probability.

LEMMA 2. *Let $\mathcal{M}_c = (S, \Sigma, \mathit{Init}_c, \mathit{Steps}_c, \mathit{UnSafe}_c)$ be an NMP and $\mathcal{M}_f = (S, \mathit{Init}_f, \mathit{Steps}_f, \mathit{UnSafe}_f)$ be a PA such that $\mathcal{M}_c \preceq_{cf} \mathcal{M}_f$. For all $s \in S$ and all non-negative n, we have*

$$\mathit{Reach}^{\mathcal{M}_c}_{\leq n}(s) \leq \mathit{Reach}^{\mathcal{M}_f}_{\leq n}(s)$$

and

$$\mathit{Reach}^{\mathcal{M}_c}(s) \leq \mathit{Reach}^{\mathcal{M}_f}(s).$$

Thus, if we can show that reachability probabilities in $\mathcal{M}_f$ are below a certain threshold, this is also the case in $\mathcal{M}_c$.

3. STOCHASTIC HYBRID AUTOMATA

In this section, we provide definitions for the fragment of stochastic hybrid automata addressed in this paper. We define the underlying semantics of these high-level models in terms of the models of Section 2 and show that the measurability of the semantics of a complete automaton follows from the measurability of its constituent parts.

3.1 Model Description

Probabilistic hybrid automata as considered in our previous work [23] require probability measures to have finite support and also require that only a finite number of non-deterministic choices occurs in each state. In the following, we describe an extension to *stochastic* hybrid automata, in which we allow continuous probability distributions and uncountable non-determinism in discrete assignments, yet not over continuous distributions.

Let m denote a variable ranging over a finite set of modes $\mathtt{M} := \{m_1, \ldots, m_n\}$, and let $\boldsymbol{x} := (x_1, \ldots, x_k)$ be a vector of variables ranging over real numbers $\mathbb{R}$. For denoting the derivatives of $\boldsymbol{x}$ we use $\dot{\boldsymbol{x}} := (\dot{x}_1, \ldots, \dot{x}_k)$, ranging over $\mathbb{R}$ correspondingly. With m' and $\boldsymbol{x}' := (x'_1, \ldots, x'_k)$ we denote primed versions of m and $\boldsymbol{x}$ respectively, as subsequently used to specify values resulting from discrete jumps of a hybrid automaton.

Later on, $S := \mathtt{M} \times \mathbb{R}^k$ will denote the state-space of the semantics of the hybrid automaton. We let $\Sigma := \mathcal{B}(S)$ denote the Borel σ-algebra on the state-space. Further, let $H(\Sigma)$ denote the σ-algebra generated by all $H_A := \{B \in \Sigma \mid A \cap B \neq \emptyset\}$ where $A \in \Sigma$. (This construction is similar to the ones used in non-probabilistic NLMP [10].) A *state-space constraint* is a constraint $\mathtt{s} \subseteq \mathtt{M} \times \mathbb{R}^k$ over modes and variables. A *flow constraint* is a constraint $\mathtt{f} \subseteq \mathtt{M} \times \mathbb{R}^k \times \mathbb{R}^k$ over the variables m, $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$.

A *probabilistic guarded command* $\mathtt{c}$ shall be defined as

$$\mathit{condition} \to p_1 : \mathit{update}_1 + \ldots + p_n : \mathit{update}_n$$

where $n \geq 1$ denotes the cardinality of the probabilistic branching of $\mathtt{c}$ with $p_i > 0$ for $i = 1, \ldots, n$ and $\sum_{i=1}^n p_i = 1$. We demand that $\mathit{condition} \in \Sigma$ is a measurable constraint over $(m, \boldsymbol{x})$, and that $\mathit{update}_i : (S, \Sigma) \to (\Sigma, H(\Sigma))$ is a measurable function denoting a reset mapping for m and $\boldsymbol{x}$ for all $i = 1, \ldots, n$. Observe that for different $i \neq j$, it could be the case that $\mathit{update}_i(m, \boldsymbol{x}) \cap \mathit{update}_j(m, \boldsymbol{x}) \neq \emptyset$. In our notation, if we do not mention a variable in a guarded command, it remains unchanged.

EXAMPLE 2.

$$\begin{aligned} m = m_1 \quad \to \quad & 0.2 : m' = m_2 \wedge x'_1 \leq x_2 - 0.84 \\ + \quad & 0.2 : m' = m_2 \wedge x_2 - 0.85 \leq x'_1 \leq x_2 - 0.25 \\ + \quad & 0.2 : m' = m_2 \wedge x_2 - 0.26 \leq x'_1 \leq x_2 + 0.26 \\ + \quad & 0.2 : m' = m_2 \wedge x_2 + 0.25 \leq x'_1 \leq x_2 + 0.85 \\ + \quad & 0.2 : m' = m_2 \wedge x'_1 \geq x_2 + 0.84 \end{aligned}$$

is a probabilistic guarded command. It can be executed when being in mode m_1. With probability 1, we move to mode m_2. With a probability of 0.2 each, a certain interval is chosen, and the variable x_1 is non-deterministically set to an arbitrary value within this interval. The endpoints of the intervals depend on the value of x_2. Other variables remain unchanged.

While previously [23] we restricted to commands where the updates $update_i$ map a state to a unique successor, we here allow the updates to be predicates over successor states. This leads to a possibly uncountable non-determinism, as in Example 2.

To model continuous measures, we introduce an additional form of guarded commands. Let $M : (S, \Sigma) \to (\Delta(S), \Delta(\Sigma))$ be a measurable function mapping states to probability measures. A *stochastic guarded command* is of the form

$$condition \to M.$$

EXAMPLE 3. *We specify $M(m_1, x_1, x_2, \ldots, x_n)$ as*

$$M(m_1, x_1, x_2, \ldots, x_n)\left(\{m_2\} \times [a, b] \times \bigtimes_{i=2}^{n} \{x_i\}\right)$$
$$:= \quad \frac{1}{\sqrt{2\pi}} \int_a^b \exp\left(-\frac{1}{2}(x - x_2)^2\right) \mathrm{d}x,$$

with the unique extension of this measure to other Borel sets. Then $m = m_1 \to M$ is a stochastic guarded command, which we denote by $\mathtt{c}$. It can execute under the same conditions as the probabilistic guarded command from Example 2, and has the same target mode. However, x_1 is set according to the normal distribution $\mathcal{N}(x_2, 1)$ with expected value x_2 and standard deviation 1. Due to the properties of the normal distribution, such a command is suited to set a variable according to a probability distribution which is centred around an ideal value from which the variable may deviate into both directions. In practice, such perturbations arise from inexact measurements, deviations of production parameters in a production line, etc.

With these preparations, we can define stochastic hybrid automata.

DEFINITION 3. *A* stochastic hybrid automaton *is a tuple $\mathcal{H} = (\mathtt{M}, \boldsymbol{x}, Init, Flow, \mathtt{C}, UnSafe)$ where*

- *$\mathtt{M}$ is a finite set of modes and $\boldsymbol{x}$ is a set of k variables,*
- *$Init \subseteq \mathtt{M} \times \mathbb{R}^k$ is a constraint on the initial states,*
- *$UnSafe \subseteq \mathtt{M} \times \mathbb{R}^k$ describes the unsafe states,*
- *$Flow \subseteq \mathtt{M} \times \mathbb{R}^k \times \mathbb{R}^k$ is a flow constraint and*
- *$\mathtt{C}$ denotes a finite set of guarded commands. We denote the subset of probabilistic guarded commands as $\mathtt{C}_f$ and the subset of stochastic guarded commands as $\mathtt{C}_c$.*

We require Flow to be measurable in the following sense: for each $m \in \mathtt{M}$, the pre-post-relation $T :=$

$$\left\{ (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^k \times \mathbb{R}^k \;\middle|\; \begin{array}{l} \exists e \geq 0, f : [0, e] \to \mathbb{R}^k \text{ differentiable} : \\ \left(\begin{array}{l} f(0) = \boldsymbol{x} \\ \wedge\, f(e) = \boldsymbol{y} \\ \wedge\, \forall t \in [0, e] : \\ \quad (m, f(t), \dot{f}(t)) \in Flow \end{array} \right) \end{array} \right\}$$

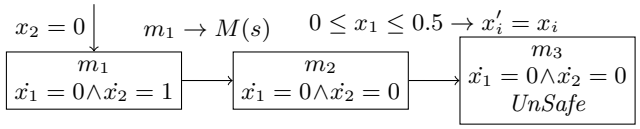


Figure 1: Example stochastic hybrid automaton. Here, $M(s)$ with $s = (m, x_1, x_2)$ is a Dirac distribution with respect to x_2 and a normal distribution over x_1, combined such that x_2 keeps its value while x_1 has expected value x_2 with standard deviation 1.

mediated by the continuous flow is a measurable set in $\mathcal{B}(\mathbb{R}^k \times \mathbb{R}^k)$, and we require $post^m(x) := T_{|\boldsymbol{x}}$ to be measurable, i.e. $post^m : (\mathbb{R}^k, \mathcal{B}(\mathbb{R}^k)) \to (\mathcal{B}(\mathbb{R}^k), H(\mathcal{B}(\mathbb{R}^k)))$. Furthermore, we require Init and UnSafe to be measurable sets.

Measurability of most of the above model constituents can be guaranteed by considering o-minimal definable sets. General results connecting o-minimal definability with measurability [6, 7] show that a sufficient criterion for the above pre-post relation T being Borel-measurable is that it is definable in some o-minimal theory over the reals. In practice, this holds for the pre-post relations manipulated by hybrid model checkers, as all current hybrid model checkers tackle differential equations by providing descriptions or approximations of their reach set and thus of the above pre-post relation via sets definable in o-minimal theories over the reals, such as finite unions of rectangular boxes, zonotopes, polyhedra, ellipsoids, or by differential invariants. In a nutshell, the general results connecting o-minimality with measurability considers the standard parts [6] of o-minimal theories and shows them to be Borel measurable. This, together with the fact that the standard part $\mathrm{st}(A)$ satisfies $\mathrm{st}(A) = A$ provided $A \subseteq \mathbb{R}^k$ [7] implies that relations definable by o-minimal theories over the reals are Borel measurable [6]. Hence, T is Borel measurable if described in some o-minimal theory. The function $post^m$ [16] is defined as section of relation T. Although this definition ensures that $post^m(x)$ gives measurable sets for every x, it does not imply measurability of $post^m$ itself, which is why we need to require it.

We talk of a *probabilistic* hybrid automaton if $\mathtt{C}_c = \emptyset$. We still allow uncountable non-determinism for this model class. In this subclass, measurability restrictions are not needed to guarantee that probability measures are well-defined in the low-level semantical model.

EXAMPLE 4. *Consider the model in Figure 1. We assume that μ is the normal distribution of the command $\mathtt{c}$ from Example 3. We are interested in the maximal probability to finally reach the target mode m_3. To obtain the maximal probability, we wait in m_1 until $x_2 = 0.25$. Then, we jump to m_2. Let $f(x)$ be the density of $\mathcal{N}(0.25, 1)$. With probability $\int_0^{0.5} f(x)\,\mathrm{d}x \approx 0.197$ we can finally jump from m_2 to m_3.*

3.2 Semantics of Stochastic Hybrid Automata

The semantics of a stochastic hybrid automaton $\mathcal{H} = (\mathtt{M}, \boldsymbol{x}, Init, Flow, \mathtt{C}, UnSafe)$ is the tuple $Sem(\mathcal{H}) := (S, \Sigma, Init, Steps, UnSafe)$ where $S := \mathtt{M} \times \mathbb{R}^k$, $\Sigma := \mathcal{B}(S)$ and we define *Steps* as the union of two transition relations $Steps_{\mathcal{T}}, Steps_{\mathcal{J}} : S \to \Delta(\Sigma)$. The semantics of timed steps is defined as

$$Steps_{\mathcal{T}}((m, \boldsymbol{x})) := \{\delta_{(m, \boldsymbol{x}')} \mid \boldsymbol{x}' \in post^m(\boldsymbol{x})\}.$$

Now we define the semantics of guarded commands. To start with, we define the semantics of a probabilistic guarded command $\mathsf{c} = con \to p_1 : u_1 + \ldots + p_n : u_n$ by: $Steps_{\mathsf{c}}(s) := \emptyset$ if $s \notin con$, and otherwise

$$Steps_{\mathsf{c}}(s) := \left\{ \sum_{i=1}^{n} p_i \delta_{s_i} \,|\, (s_1, \ldots, s_n) \in u_1(s) \times \cdots \times u_n(s) \right\}.$$

Inside the above formula, we have weighted sums of Dirac probability measures. A step induced by a probabilistic guarded command has as successors all measures, such that with probability p_i we choose a state of the ith update. For measures in which we have a state which is the successor of different updates, the probabilities of these updates are added up.

Next, for a stochastic guarded command $\mathsf{c} = con \to M$ we define

$$Steps_{\mathsf{c}}(s) := \begin{cases} \{M(s)\} & \text{if } s \in con, \\ \emptyset & \text{else.} \end{cases}$$

Then for $s \in S$, we let

$$Steps_{\mathcal{J}}(s) := \bigcup_{\mathsf{c} \in \mathbb{C}} Steps_{\mathsf{c}}(s)$$

and

$$Steps(s) := \begin{cases} Steps_{\mathcal{T}}(s) \cup Steps_{\mathcal{J}}(s), \\ \qquad \text{if } Steps_{\mathcal{T}}(s) \cup Steps_{\mathcal{J}}(s) \neq \emptyset, \\ \{\delta_s\} \text{ else.} \end{cases}$$

The possible steps in the semantics are thus all possible transitions induced by jumps or timed transitions. The self-loops introduced using Dirac distributions are necessary to guarantee that each state has at least one successor measure.

It remains to show that the semantics is well-defined. Because of this, for $s \in S$ it must be the case that $Steps(s)$ is an element of $\Delta(\Sigma)$ and that $Steps$ is a measurable function. If this holds, then $Sem(\mathcal{H})$ is indeed an NMP.

LEMMA 3. *Let $Sem(\mathcal{H}) = (S, \Sigma, Init, Steps, UnSafe)$ be a tuple that is the semantics of a stochastic hybrid automaton $\mathcal{H}$. Then, $Steps : (S, \Sigma) \to (\Delta(\Sigma), H(\Delta(\Sigma)))$ is a measurable function mapping states to elements of $\Delta(\Sigma)$. In turn, $Sem(\mathcal{H})$ is an NMP. In case $\mathcal{H}$ is purely probabilistic, $Sem(\mathcal{H})$ is a PA.*

4. OVER-APPROXIMATING STOCHASTIC HYBRID AUTOMATA

Over-approximation of the semantics of a *probabilistic* hybrid automaton $\mathcal{H}$ by a finite PA $\mathcal{M}$ with an abstract state-space, written $\mathcal{M} \in Abs_f(\mathcal{H})$, implies that the reachability probability of unsafe states in $\mathcal{M}$ is no lower than in $\mathcal{H}$ [22, 23].

Now we describe how we can over-approximate a *stochastic* hybrid automaton by a *probabilistic* hybrid automaton. At first, we describe how to abstract a single stochastic guarded command into a probabilistic command.

DEFINITION 4. *Consider a stochastic guarded command c defined by condition $\to M$. Fix $p_i \in [0, 1]$ such that $\sum_{i=1}^{n} p_i = 1$. Let $\hat{g}_1, \ldots, \hat{g}_n : S \to \Sigma$ be functions such that $M(s)(\hat{g}_i(s)) = p_i$, and $M(s)\left(\bigcup_{i=1}^{n} \hat{g}_i(s)\right) = 1$, for all $s \in S$. Further, we require the sets $\hat{g}_1(s), \ldots, \hat{g}_n(s)$ to be pairwise disjoint. Let $g_1, \ldots, g_n : S \to 2^S$ be functions satisfying $\hat{g}_i(s) \subseteq g_i(s)$, for all $s \in S$. We define the probabilistic guarded command $Abs_c(\mathsf{c}, g_1, \ldots, g_n, p_1, \ldots, p_n)$ as*

$$condition \to p_1 : g_1 + \ldots + p_n : g_n,$$

and call g_i abstraction functions.

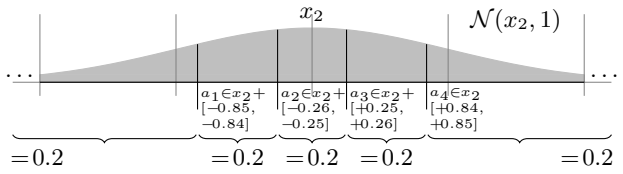


Figure 2: Normal distribution for Example 5.

By abstracting a command this way, we may introduce uncountable additional non-determinism. Overlapping sets $g_i(s), g_j(s)$, $i \neq j$ are allowed. This feature can be used for instance, if the exact $\hat{g}_i(s)$, $\hat{g}_j(s)$ corresponding to probabilities p_i, p_j cannot be computed. As already stated in Section 2, this is no drawback of our method. In the final abstraction that we compute, the non-determinism will be over-approximated by a finite number of transitions. Notice that the abstraction of a single command is done symbolically in the high-level description of the probabilistic hybrid automaton instead of the low-level model, as in grid-based methods.

EXAMPLE 5. *Consider the stochastic guarded command from Example 3. Let $p_1 := \ldots := p_5 := 0.2$ and consider $a_1 \in x_2 + [-0.85, -0.84]$, $a_2 \in x_2 + [-0.26, -0.25]$, $a_3 \in x_2 + [0.25, 0.26]$, $a_4 \in x_2 + [0.84, 0.85]$. We define*

$$\begin{aligned} I_1 &:= \{m_2\} \times (-\infty, a_1] \times \bigtimes_{i=2}^{n} \{x_i\}, \\ I_2 &:= \{m_2\} \times (a_1, a_2] \times \bigtimes_{i=2}^{n} \{x_i\}, \\ &\ldots. \end{aligned}$$

Assume that by a precomputation (which will later on be described in Subsection 4.1) we known that for states $s = (m_1, x_1, \ldots, x_n)$ it is

$$M(s)(I_1) = M(s)(I_2) = M(s)(I_3) = \ldots = p_i = 0.2,$$

as illustrated in Figure 2. For each point s' of the support of $M(s)$, we have at least one I_i which contains s'. We also know that $M(s)(\{m_2\} \times (-\infty, \infty) \times \bigtimes_{i=2}^{n} \{x_i\}) = 1$. Thus, we can define

$$\begin{aligned} \hat{g}_1(s) &:= \{m_2\} \times (-\infty, a_1] \times \bigtimes_{i=2}^{n} \{x_i\} \\ \subseteq g_1(s) &:= \{m_2\} \times (-\infty, x_2 - 0.84] \times \bigtimes_{i=2}^{n} \{x_i\}, \\ \hat{g}_2(s) &:= \{m_2\} \times (a_1, a_2] \times \bigtimes_{i=2}^{n} \{x_i\} \\ \subseteq g_2(s) &:= \{m_2\} \times [x_2 - 0.85, x_2 - 0.25] \times \bigtimes_{i=2}^{n} \{x_i\}, \\ &\ldots. \end{aligned}$$

Because of this, the probabilistic guarded command of Example 2 is an abstraction of the stochastic guarded command of Example 3.

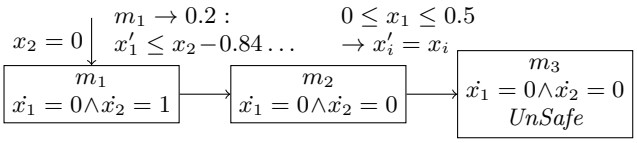


Figure 3: Probabilistic hybrid automaton abstraction of the stochastic hybrid automaton of Figure 1.

In the abstraction of a complete stochastic hybrid automaton, all stochastic guarded commands are abstracted by probabilistic guarded commands.

DEFINITION 5. *Let $\mathcal{H} = (\mathtt{M}, \boldsymbol{x}, Init, Flow, \mathcal{C}, UnSafe)$ be a stochastic hybrid automaton, and consider a family of abstraction functions $F = ((g_{\mathtt{c},1}, \ldots, g_{\mathtt{c},n}), (p_{\mathtt{c},1}, \ldots, p_{\mathtt{c},n}))_{\mathtt{c} \in \mathcal{C}_c}$ with corresponding probabilities. Then we define the* probabilistic hybrid automaton abstraction *of $\mathcal{H}$ as*

$$Abs_c(\mathcal{H}, F) := (\mathtt{M}, \boldsymbol{x}, Init, Flow, Abs_c(\mathcal{C}), UnSafe)$$

where $Abs_c(\mathcal{C}) := \mathcal{C}_f \cup \{Abs_c(\mathtt{c}, g_{\mathtt{c},1}, \ldots, g_{\mathtt{c},n}, p_{\mathtt{c},1}, \ldots, p_{\mathtt{c},n}) \mid \mathtt{c} \in \mathcal{C}_c\}$.

We state the correctness of the over-approximation.

LEMMA 4. *Consider a stochastic hybrid automaton $\mathcal{H}$ and a family F of abstraction functions with corresponding probabilities. Then $Sem(\mathcal{H}) \preceq_{cf} Sem(Abs_c(\mathcal{H}, F))$.*

EXAMPLE 6. *In Figure 3 we give a possible over-approximation of the automaton of Figure 1. We use the abstraction from Example 2 (denoted $\mathtt{c}'$) of the stochastic guarded command of Example 3 (denoted by $\mathtt{c}$) (see Example 5) for the only stochastic guarded command in this model. In the abstraction, the reachability probability is higher than it was originally. If in m_1 we wait until $x_2 = 0.2$, for two branches of $\mathtt{c}'$ we may enter m_3, thus the reachability probability is $2 \cdot 0.2 = 0.4$. By splitting M into more equidistant parts, we could decrease this probability.*

4.1 Obtaining Abstraction Functions

For the abstraction to be applicable in practice, it is crucial to compute the family of abstraction functions. As there exist quite diverse forms of random variables, we cannot give an algorithm to handle all cases. Instead, we sketch how to obtain over-approximation functions for certain classes of random variables.

At first, consider a probability measure $\mu : \mathcal{B}(\mathbb{R}) \to [0, 1]$ given by a density function $f(x)$, for instance the normal distribution. Using numerical methods, we can then compute bounds for a_i such that $\mu((-\infty, a_1]) = p_1, \mu((a_1, a_2]) = p_2, \ldots, \mu((a_{n-1}, \infty)) = p_n$, for some fixed $p_1, \ldots, p_n$. Following Example 5, for $\mathcal{N}(0, 1)$ and $n = 5, p_i = 0.2$, we could obtain $a_1 \in [-0.85, -0.84], a_2 \in [-0.26, -0.25], \ldots$. We transform the random variable to get state-dependent intervals. For the example, we use that $\mathcal{N}(x, y) = \frac{\mathcal{N}(0,1) - x}{y}$. Thus, we can transform corresponding interval endpoints b_i to $b_i(x, y) = b_i(0, 1) \cdot y + x$. When setting $x = x_2, y = 1$, we obtain the same intervals as given in Example 5.

If the cumulative distribution function $F(x)$ of a random variable is known and we can compute a closed-form of F^{-1}, we can use a method similar to the inverse transform method. Consider the exponential distribution with state-dependent λ. We have that if $F_\lambda(a_i) = p_i$ then $a_i = -\ln(1 - p_i)\frac{1}{\lambda}$. We can then obtain adjoint intervals which have a certain probability by precomputing $[b_i, b_i'] \ni -\ln(1 - p_i)$ and thus we specify command branches $p_i : \frac{b_{i-1}}{\lambda} \leq x \leq \frac{b_i'}{\lambda}$.

For probability measures in two variables, we consider $f(\cdot, (-\infty, \infty))$ first, and then split each $f([a_i, a_{i+1}], \cdot)$ again. This technique extends to any finite number k of variables. If we split each of them into a number of n parts, the support of the abstracting distribution has a size of n^k. Thus, the worst-case complexity of this method is rather bad. However, the case that only one or few variables change appears to be the practically relevant case for us. It occurs in settings where the environment can be observed only with limited accuracy, as the ones discussed in Section 5.

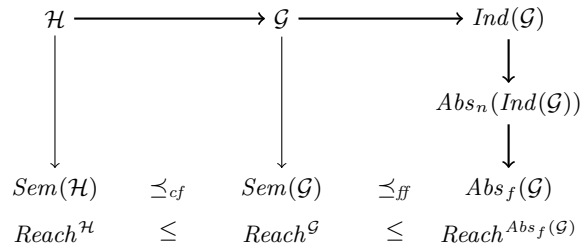


Figure 4: Scheme of abstraction. $\mathcal{G}$ denotes the abstraction $Abs_c(\mathcal{H}, F)$ and $Abs_n(.)$ denotes abstraction of a non-probabilistic hybrid automata, and $\preceq_{ff}$ denotes the simulation relation on PAs [21].

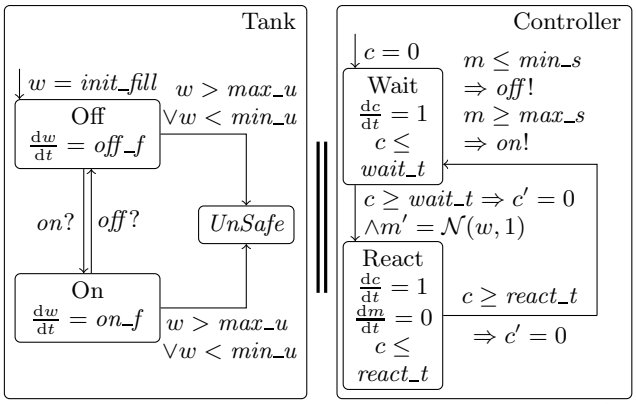
4.2 Finite Abstractions

In the previous sections, we have seen that a stochastic hybrid automaton can be abstracted by a probabilistic hybrid automaton with uncountably many states and transitions. To effectively obtain probability bounds, we abstract this automaton by a finite state probabilistic automaton, which can then be submitted to a probabilistic model checker for further analysis. We can do so by harvesting previous work [23], which proceeds via a non-probabilistic version $Ind(\mathcal{G})$ of the original probabilistic hybrid automaton $\mathcal{G}$ to arrive at a finite abstraction $Abs_f(\mathcal{G})$. A slight adaptation is needed since we thus far did not consider non-determinism within one command, which we now require in order to over-approximate stochastic guarded commands. Nevertheless, the correctness proofs stay unchanged when allowing it. Due to space limitations, we cannot give a complete description of the previous work here.

An overview of the entire approach is depicted in Figure 4. The computation of maximal reachability probabilities in the resulting finite-state PA is done via well-established numerical recipes, and is the capstone ingredient in this effective computation of safe upper bounds for reachability properties for this general class of stochastic hybrid automata.

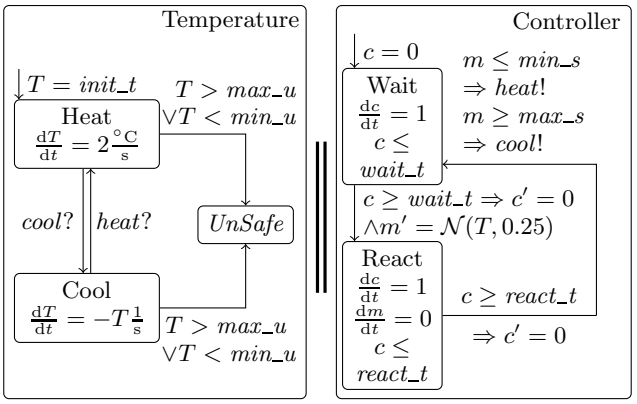
5. EXPERIMENTS

We experiment with the approach outlined thus far using the tool PROHVER (probabilistic hybrid automata verifier) [23] on a selection of stochastic hybrid automata case studies. In each case, we first abstract stochastic guarded commands (so far manually) to probabilistic guarded commands, which PROHVER can handle. Thus, we obtain a



$init\_fill = 6.5\text{l}$, $wait\_t = 1\text{s}$, $react\_t = 0.1\text{s}$, $off\_f = 1\frac{\text{l}}{\text{s}}$, $on\_f = -2\frac{\text{l}}{\text{s}}$, $max\_s = 8\text{l}$, $min\_s = 5\text{l}$, $max\_u = 12\text{l}$, $min\_u = 1\text{l}$

Figure 5: Water level control with perturbed measurements, modelling measurement deviation by normal distribution $\mathcal{N}(w, 1)$.



$init\_t = 8^\circ\text{C}$, $wait\_t = 1\text{s}$, $react\_t = 0.1\text{s}$, $max\_s = 9^\circ\text{C}$, $min\_s = 6^\circ\text{C}$, $max\_u = 12^\circ\text{C}$, $min\_u = 3^\circ\text{C}$

Figure 6: Temperature control with perturbed measurements, a variant of the model from Figure 5 exhibiting more complex continuous dynamics.

probabilistic hybrid automaton which over-approximates the original stochastic hybrid automaton. Then, our tool uses a modified version of PHAVer [13] to obtain the transition relation of a finite-state abstraction of a non-probabilistic projection of the hybrid automaton. ProHVer then reintroduces the probabilities to this abstraction and constructs a corresponding finite-state Markov decision process. The maximal reachability probabilities we can obtain herein over-approximate the ones which can be obtained in the semantics of the original stochastic hybrid automaton.

To show the applicability of our approach, we applied ProHVer on three case studies, which are small but diverse in the nature of their behaviour. In the examples considered, we focus on reachability probabilities with upper time bounds (obtained by using an additional clock to measure time), because these correspond to very natural verification problems for the settings considered. Notably, our method is not restricted to time-bounded reachability. Actually, time-unbounded problems are simpler (because no additional clock is needed). In the examples considered, time-unbounded reachability probabilities would always be 1. Experiments were run on a Pentium 4 with 2.67 GHz and 4 GB RAM. Models and tools can be found on `http://depend.cs.uni-saarland.de/tools/prohver/`.

5.1 Water Level Control

We consider a model of a water level control system (extended from the one of Alur et al. [4] and our previous paper [23]). In particular, we use this case study to demonstrate the influence which different abstractions of the same continuous stochastic command have. A water tank is filled by a constant stream of water, and is connected to a pump which is used to avoid overflow of the tank. A control system operates the pump in order to keep the water level within predefined bounds. The controller is connected to a sensor measuring the level of water in the tank. A sketch of the model is given in Figure 5. The state "Tank" models the tank and the pump, and w is the water level. Initially, the tank contains a given amount of water. Whenever the pump is turned off in state "Off", the tank fills with a constant rate due to the inflow. Conversely, more water is pumped out than flows in when the pump is on.

The controller is modelled by automaton "Controller". In state "Wait", the controller waits for a certain amount of time. Upon the transition to "React", the controller measures the water level. To model the uncertainties in measurement, we set the variable m to a normal distribution with expected value w (the actual water level) and standard deviation 1. According to the measurement obtained, the controller switches the pump off or on.

We are interested in the probability that within a given time bound, the water level leaves the legal interval. In Table 1, we give upper bounds for this probability for different time bounds as well as the number of states in the abstraction computed by PHAVer and the time needed for the analysis. For the stochastic guarded command simulating the measurement, we consider different abstractions by probabilistic guarded commands of different precision, for which we give the abstraction functions in the table caption. When we refine the abstraction A to a more precise B, the probability bound decreases. If we introduce additional non-determinism as in abstraction C, probabilities increase again. If we refine B again into D, we obtain even lower probability bounds. The price to be paid for increasing precision, however, is in the number of abstract states computed by PHAVer as well as a corresponding increase in the time needed to compute the abstraction.

Manual analysis shows that in this case study, the over-approximation of the probabilities only results from the abstraction of the stochastic guarded command into a probabilistic guarded command and is not increased further by the state-space abstraction.

5.2 Temperature Control

We consider a temperature control system extended from a previous case study [23], originally studied by Alur et al. [5]. In Figure 6 we depict the system structure. We ask whether using an air conditioning control system we are able to keep the temperature of a room within a certain range. In contrast to the water level case, the model features dynamics governed by linear rather than piecewise constant ODE, and

time bound	Abstraction A			Abstraction B			Abstraction C			Abstraction D		
	prob.	build (s)	states	prob.	build (s)	states	prob.	build (s)	states	prob.	build (s)	states
20s	0.1987	3	999	0.0982	3	1306	0.1359	3	1306	0.0465	5	1920
30s	0.2830	6	2232	0.1433	8	2935	0.1870	8	2935	0.0693	15	4341
40s	0.3580	16	3951	0.1860	18	5212	0.2547	18	5212	0.0916	47	7734
50s	0.4250	34	6156	0.2264	42	8137	0.3024	43	8137	0.1134	108	12099
60s	0.4848	67	8847	0.2647	86	11710	0.3577	85	11710	0.1347	219	17436

Table 1: Water level control results. We round probabilities to four decimal places. Abstractions used are $A = w + \{[-2,2],(-\infty,1.9] \cup [1.9,\infty)\}$, $B = w + \{[-2,2],(-\infty,1.9],[1.9,\infty)\}$, $C = w + \{[-2.7,2.7],(-\infty,1.2),[1.2,\infty)\}$, $D = w + \{[-1.5,1.5],[-1.5,-2],[1.5,2],(-\infty,1.9),[1.9,\infty)\}$.

instead of varying the splitting of the normal distribution, we vary the refine interval used by PHAVer to analyse such systems. Smaller intervals lead to more precise abstractions.

In Table 2, we give probability bounds and performance statistics. We used a refine interval on the variable T which models the temperature. The interval lengths are given in the table. For all instances there is an interval length small enough to obtain a probability bound that is the best possible using the given abstraction of the normal distribution. Smaller intervals were of no use in this case. The drastic discontinuities in probability bounds obtained are a consequence abstraction by PHAVer.

5.3 Moving-block Train Control

As a more complex example of a hybrid system implementing a safety-critical control policy, we present a model of headway control in the railway domain (Figure 7). A more extensive description of the setting plus a closely related case study containing a sampling-related bug not present in the current model appeared in a different publication [17]. In contrast to fully automated transport, which is in general simpler to analyse (as the system is completely under control of the embedded systems) our sample system implements safe-guarding technology that leaves trains under full human control provided safety is not at risk. It is thus an open system, giving rise to the aforementioned analysis problems.

Our model implements safe interlocking of railway track segments by means of a "moving block" principle of operation. While conventional interlocking schemes in the railway domain lock a number of static track segments in full, the moving block principle enhances traffic density by reserving a "moving block" ahead of the train which moves smoothly with the train. This block is large enough to guarantee safety even in cases requiring emergency stops, i.e. has a dynamically changing block-length depending on current speed and braking capabilities. There are two variants of this principle, namely train separation in relative braking distance, where the spacing of two successive trains depends on the current speeds of both trains, and train separation in absolute braking distance, where the distance between two trains equals the braking distance of the second train plus an additional safety distance (here given as $sd = 400$m). We use the second variant, as also employed in the European Train Control System (ETCS) Level 3.

Our simplified model consists of a leader train, a follower train, and a moving-block control regularly measuring the leader train position and communicating a related *movement authority* to the follower. The leader train is freely controlled by its operator within the physical limits of the train, while the follower train may be forced to controlled braking if coming close to the leader. The control principle is as follows:

1. 8 seconds after communicating the last movement authority, the moving-block control takes a fresh measurement m of the leader train position s_l. This measurement may be noisy.

2. Afterwards, a fresh movement authority derived from this measurement is sent to the follower. The movement authority is the measured position m minus the length l of the leader train and further reduced by the safety distance sd. Due to an unreliable communication medium, this value may reach the follower (in which case its movement authority $auth$ is updated to $m - l - sd$) or not. In the latter case, which occurs with probability 0.1, the follower's movement authority stays as is.

3. Based on the movement authority, the follower continuously checks the deceleration required to stop exactly at the movement authority. Due to PHAVer being confined to linear arithmetic, this deceleration is conservatively approximated as $a_{\text{req}} = \frac{v \cdot v_{\max}}{2(s-auth)}$, where v is the actual speed, $v_{\max}$ the (constant) top speed, and s the current position of the follower train, rather than the physically more adequate, yet non-linear, $a_{\text{req}} = \frac{v^2}{2(s-auth)}$ of the original model [17].

4. The follower applies automatic braking whenever the value of a_{req} falls below a certain threshold b_{on}. In this case, the follower's brake controller applies maximum deceleration $a_{\min}$, leading to a stop before the movement authority as $a_{\min} < b_{\text{on}}$. Automatic braking ends as soon as the necessary deceleration a_{req} exceeds a switch-off threshold $b_{\text{off}} > b_{\text{on}}$. The thresholds b_{on} and b_{off} are separate to prevent the automatic braking system from repeatedly engaging and disengaging in intervals of approximately 8 seconds when the leading train is moving.

We consider the probability to reach the state "Crash" in which the follower train has collided with the leader train. In Table 3, we give probability bounds and performance results. We modelled the measurement error using a normal distribution with expected value s_l, i.e. the current position of the leader train. In the table, we considered different standard deviations of the measurement. The abstraction used for each of them can be obtained using structurally equal Markov decision processes, only with different probabilities. Thus, we only needed to compute the abstraction

time bound	interval length ∞			interval length 2			interval length 1			interval length 0.5		
	prob.	build (s)	states	prob.	build (s)	states	prob.	build (s)	states	prob.	build (s)	states
2s	1	0.03	7	0	0.17	16	0	0.21	21	0	0.30	31
4s	1	0.05	23	1	1.26	269	0.284643	1.61	316	0.284643	2.97	546
6s	1	0.07	39	1	5.79	1518	0.360221	8.66	2233	0.360221	17.39	3797
8s	1	0.10	55	1	19.27	4655	1	35.62	8261	0.488265	81.39	16051
10s	1	0.12	71	1	53.25	10442	1	119.33	20578	0.590683	507.12	44233

Table 2: Temperature control results. To abstract $\mathcal{N}(T, 0.25)$, we used $T + \{[-0.25, 0.25], (-\infty, -0.25], [0.25, \infty)\}$.

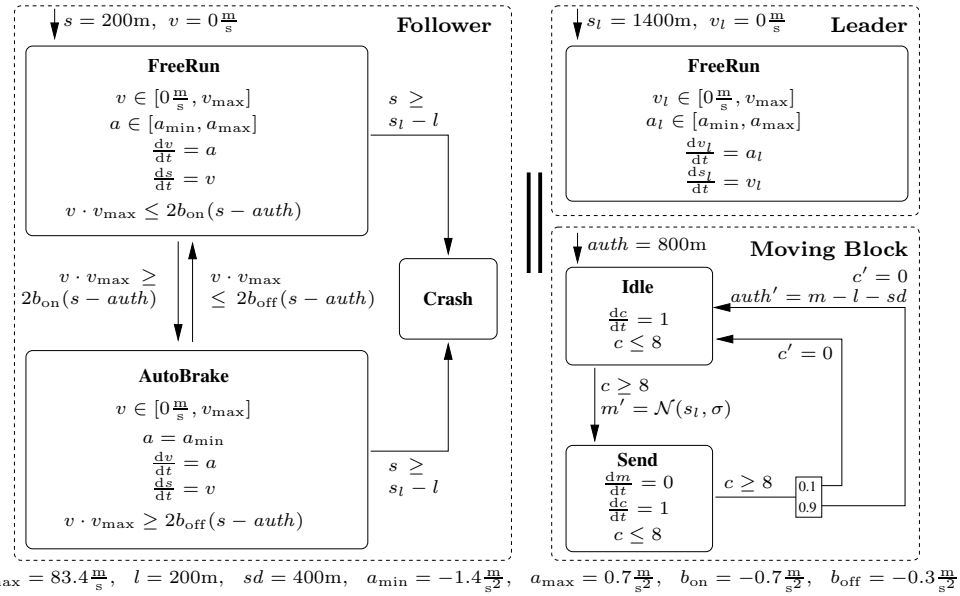


Figure 7: Moving-block train distance control with perturbed measurement of leader train position (using normal distribution $\mathcal{N}(s_l, \sigma)$ centred around actual value, with standard deviation σ) and unreliable communication of resultant movement authorities (failure probability 0.1). "Crash" represents collision of trains.

once for all deviations, and just had to change the transition probabilities before obtaining probability bounds from the abstraction. It was sufficient to split the normal distribution into two parts. Depending on where we set the split-point, we obtained probability bounds of different quality. Although this hybrid automaton is not piecewise constant, such that PHAVer needs to over-approximate the set of reachable states, we are still able to obtain useful probability bounds when using an adequate abstraction, without refine intervals.

The graph in Figure 8 reveals the expected positive correlation between measurement error and risk, but also the effectiveness of the fault-tolerance mechanism handling communication loss. We see that crashes due to communication losses are effectively avoided, rooted in the principle of maintaining the last received movement authority whenever no fresh authority is at hand. In fact, risk correlates negatively with the likelihood of communication loss. The function correlating risk to measurement error and probability of communication loss has been computed by the tool Param [15].

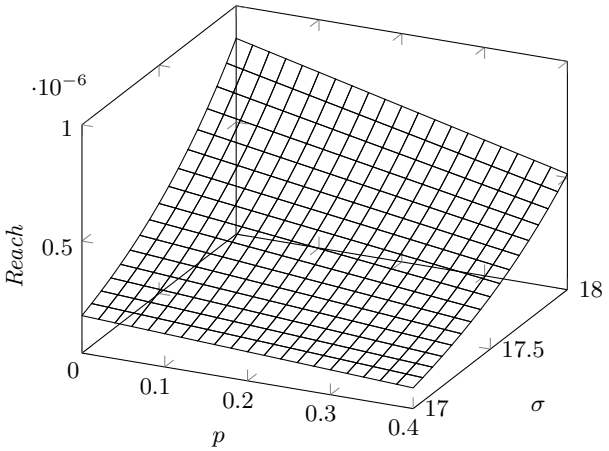


Figure 8: Bounds for probability of crash as a function of probability of movement authority loss p and standard deviation σ of distance measurement. A time bound of 100s and Abstraction A was used.

6. CONCLUSION

In this paper, we have defined a notion of stochastic hybrid systems which supports both non-determinism and continuous probability distributions in discrete jumps. We have discussed well-definedness of the semantics and have developed means to safely over-approximating reachability probabilities for such systems. As the underlying state-space abstraction which we exploit for computing probabilities is the one computed with the help of model checkers for non-stochastic hybrid systems, improvements in efficiency of such tools directly carry over to the technique we describe. The applica-

time bound	Abstraction A					Abstraction B				
	probability ($\sigma = 10, 15, 20$)			build (s)	states	probability ($\sigma = 10, 15, 20$)			build (s)	states
60s	7.110E-19	6.215E-09	2.141E-05	65	571	1.806E-06	2.700E-03	3.847E-02	62	571
80s	1.016E-18	8.879E-09	3.058E-05	201	1440	2.580E-06	3.855E-03	5.450E-02	183	1440
100s	1.219E-18	1.066E-08	3.669E-05	470	2398	3.096E-06	4.624E-03	6.504E-02	472	2392
120s	1.524E-18	1.332E-08	4.587E-05	1260	4536	3.870E-06	5.777E-03	8.063E-02	1210	4524
140s	1.727E-18	1.509E-08	5.198E-05	2541	6568	4.386E-06	6.544E-03	9.088E-02	2524	6550
160s	2.031E-18	1.776E-08	6.116E-05	5764	10701	5.160E-06	7.695E-03	1.060E-01	5700	10665

Table 3: Train control results. For abstraction A we use a division of the normal distribution into $s_l + \{(-\infty, 91], [89, \infty)\}$. For B, we split the distribution into $s_l + \{(-\infty, 51], [49, \infty)\}$. We give probabilities for different values σ of the standard deviation of the measurement.

bility of our approach has been demonstrated on three case studies, tackled using a prototypical implementation.

As future work, we want to extend our techniques to reason about the loss of precision introduced by the abstraction and consider the question how to split continuous distributions in an optimal way. We have assumed a finite number of modes and commands. Using additional abstractions [18] for the discrete part, our technique can be extended to models which have a very large or infinite number of modes.

Acknowledgements. This work was supported by the SFB/TR 14 AVACS, by ANPCyT PICT 02272, by SeCyT-UNC 2010-2011, by the VKR Centre of Excellence project MT-LAB, the DAAD-MinCyT project QTDDS, FP7-ICT MoVeS and FP7-ICT Quasimodo.

We would like to thank Pedro Sánchez Terraf from the National University of Córdoba and Stefan Ratschan from the Academy of Sciences of the Czech Republic for fruitful discussions on the measurability problems.

7. REFERENCES

[1] A. Abate, J. Katoen, J. Lygeros, and M. Prandini. Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 2010.

[2] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724–2734, 2008.

[3] E. Altman and V. Gaitsgory. Asymptotic optimization of a nonlinear hybrid system governed by a Markov decision process. *SIAM Journal of Control and Optimization*, 35(6):2070–2085, 1997.

[4] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *TCS*, 138:3–34, 1995.

[5] R. Alur, T. Dang, and F. Ivancic. Predicate abstraction for reachability analysis of hybrid systems. *ACM Transactions on Embedded Computing Systems*, 5(1):152–199, 2006.

[6] Y. Baisalov and B. Poizat. Paires de structures o-minimales. *J. Symb. Log.*, 63(2):570–578, 1998.

[7] A. Berarducci and M. Otero. An additive measure in o-minimal expansions of fields. *The Quarterly Journal of Mathematics*, 55(4):411–419, 2004.

[8] H. Blom and J. Lygeros. *Stochastic Hybrid Systems: Theory and Safety Critical Applications*, volume 337 of *LNCIS*. Springer, 2006.

[9] M. L. Bujorianu. Extended stochastic hybrid systems and their reachability problem. In *HSCC*, pages 234–249, 2004.

[10] P. R. D'Argenio, P. S. Terraf, and N. Wolovick. Bisimulations for nondeterministic labeled Markov processes. *Math. Struct. in Comp. Science*, 2010. Under consideration for publication.

[11] P. R. D'Argenio, N. Wolovick, P. S. Terraf, and P. Celayes. Nondeterministic labeled Markov processes: Bisimulations and logical characterization. In *QEST*, pages 11–20. IEEE Computer Society, 2009.

[12] M. H. A. Davis. Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society*, 46(3):353–388, 1984.

[13] G. Frehse. PHAVer: Algorithmic verification of hybrid systems past HyTech. In *HSCC*, pages 258–273, 2005.

[14] M. Giry. A categorical approach to probability theory. In *Categorical Aspects of Topology and Analysis*, pages 68–85. Springer, 1982.

[15] E. M. Hahn, H. Hermanns, B. Wachter, and L. Zhang. PARAM: A model checker for parametric Markov models. In *CAV*, pages 660–664, 2010.

[16] T. A. Henzinger. The theory of hybrid automata. In *LICS*, pages 278–292, 1996.

[17] C. Herde, A. Eggers, M. Fränzle, and T. Teige. Analysis of hybrid systems using HySAT. In *ICONS*, pages 196–201. IEEE Computer Society, 2008.

[18] M. Kwiatkowska, G. Norman, and D. Parker. A framework for verification of software with time and probabilities. In *FORMATS*, volume 6246 of *LNCS*, pages 25–45. Springer, 2010.

[19] S. Prajna, A. Jadbabaie, and G. J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE TAC*, 52(8):1415–1429, 2007.

[20] M. Prandini and J. Hu. A stochastic approximation method for reachability computations. In Blom and Lygeros [8], pages 107–139.

[21] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. *NJC*, 2(2):250–273, 1995.

[22] J. Sproston. Decidable model checking of probabilistic hybrid automata. In *FTRTFT*, pages 31–45, 2000.

[23] L. Zhang, Z. She, S. Ratschan, H. Hermanns, and E. M. Hahn. Safety verification for probabilistic hybrid systems. In *CAV*, pages 196–211, 2010.