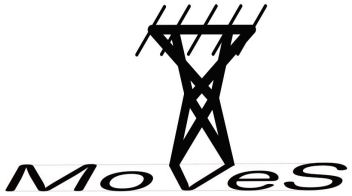


Modeling, ***Ve***rification and Control of complex ***S***ystems: From foundations to power network applications

Project co-funded by the European Community within the 7^{th} Framework Programme

---

Report D1.1
Report on Modeling Framework, Model Composition, and Bisimulation Notions

PROJECT TITLE:	Modeling, verification and control of complex systems: From foundations to power network applications	
PROJECT ACRONYM:	MoVeS	
GRANT AGREEMENT NUMBER:	257005	
PROJECT START DATE:	01.10.2010	
DURATION:	36 Months	
PROJECT COORDINATOR:	Prof. Dr. John Lygeros	
PROJECT MEMBERS:	Eidgenössische Technische Hochschule Zürich	(ETH Zurich)
	Rheinisch-Westfälische Technische Hochschule Aachen	(RWTHA)
	Technische Universiteit Delft	(TU Delft)
	Politecnico Di Milano	(PoliMi)
	OFFIS e.V.	(OE)
	Honeywell, Spol. S.R.O	(HSS)

DOCUMENT IDENTIFIER:	D1.1
ISSUE:	7.0
ISSUE DATE:	10.10.2011
PREPARED:	Technische Universiteit Delft Rheinisch-Westfälische Technische Hochschule Aachen OFFIS e.V.
APPROVED:	Eidgenössische Technische Hochschule Zürich Politecnico Di Milano Technische Universiteit Delft Rheinisch-Westfälische Technische Hochschule Aachen OFFIS e.V.

Authors

Author	Affiliation
A. Abate	TU Delft
J.P. Katoen	RWTHA
M. Fränzle	OE
M. Prandini	PoliMi

Internal Reviewers

Author	Affiliation
J. Lygeros All the authors listed above	ETH Zurich

HISTORY CHART

Issue	Date	Changed Pages	Cause of Change	Implemented by
1.0	01/08/2011	All sections	New Document	A. Abate J.P. Katoen M. Fränzle
2.0	01/09/2011	All sections	Feedback on content	M. Prandini
3.0	20/09/2011	All sections	Extensions and cross-feedback	A. Abate J.P. Katoen M. Fränzle
4.0	26/09/2011	All sections	Revision and finalization	A. Abate
5.0	05/10/2011	All sections	Additional feedback on content	M. Prandini
6.0	07/10/2011	All sections	Revision	A. Abate M. Fränzle
7.0	10/10/2011	All sections	Revision and finalization	A. Abate

Contents

1 Introduction

1.1 MoVeS

This project puts forward novel methods for modelling, analysis and control of complex, large scale systems. Fundamental research is motivated by applied problems in power networks. MoVeS adopts the framework of stochastic hybrid systems (SHS), which allows one to capture the interaction between continuous dynamics, discrete dynamics and probabilistic uncertainty. In the context of power networks, SHS arise naturally: continuous dynamics models the evolution of voltages, frequencies, etc., while discrete dynamics reflects changes in network topology, and probability represents the uncertainty about power demand and (with the advent of renewables) power supply.

More generally, because of their versatility, SHS are recognized as an ideal framework for capturing the intricacies of complex, large scale systems. Motivated by this, considerable research effort has been devoted to the development of modelling, analysis and control methods for SHS, in the computer sciences (giving rise to theorem proving and model checking methods) as well as in control engineering (giving rise to optimal control and randomized methods). Despite several success stories, however, none of the methods currently available are powerful enough to deal with real life large scale applications. A key reason for this is that the methods have been developed by different communities in relative isolation, motivated by different applications. As a consequence synergies between them have never been fully explored. MoVeS proposes to systematically exploit such synergies, by bringing together experts on all the state of the art SHS methods, will establish links between model checking, theorem proving, optimal control and randomized methods. Leveraging on their complementary strengths MoVeS will develop combined strategies and tools to enable novel applications to complex, large scale systems. Common power networks case studies will provide a testing ground for the fundamental developments, motivate them, and keep them focused.

1.2 Work Package 1

Work Package 1 (WP1) is a foundational and theoretical package, whose main goal is to establish connections between heterogeneous models of stochastic hybrid systems, to investigate and analyze model properties, and to develop general model composition techniques. Within WP1 classes of stochastic hybrid systems will be formally put in relationship and quantitatively compared through the notions of abstraction and bisimulation.

1.3 Objectives of Deliverable D1.1

Deliverable D1.1 reports on modeling frameworks, on the concept of model composition, and on notions of bisimulation relations. The report recapitulates the outcomes of Task 1.1, titled *"heterogeneous models of stochastic hybrid systems: properties and compositions"*.

Task 1.1 proposes to comparatively investigate the heterogeneous semantics and the specific descriptive capabilities of stochastic hybrid models proposed in different areas, among which systems and probability theory, stochastic processes, process algebra, discrete-event and concurrency theory.

Task 1.1 aims at elucidating relations (e.g. pre-orders, equivalences) between the different heterogeneous models, in order to generate sufficient conditions for property preservation between them. Due to the complexity of large-scale, networked, complex systems, such as power transmission networks, compositional modeling techniques are indispensable for obtaining computationally scalable analysis and control procedures. Such techniques enable a structured modeling of the system whereby the global properties can be analyzed through the independent study of system components. The goal of WP1 (and, as a first step, of Task 1.1) is to develop compositional techniques over classes of stochastic hybrid systems to address questions like: What does parallel composition mean and how can we define it? What kind of properties does it exhibit? How can hiding – an essential operator for abstraction – be defined?

This Deliverable D1.1 is structured as follows:

- Section 2 gives a review of models that embed either probability or hybrid dynamics, which are of interest for the goals of this project. As anticipated above, we cover models from systems theory (Stochastic Hybrid Systems), from probability theory (Markov Chains) and stochastic processes (Piecewise-deterministic Markov Processes, Switching Diffusions) from formal verification (Probabilistic Hybrid Automata), from process algebra (Probabilistic Automata), from discrete-event systems (Stochastic Max-Plus Models) and from concurrency theory (Interactive Markov Chains). The Section will conclude with a discussion on the structural relationships between the presented models.

- Section 3 covers the topic of model composition. It reviews classical notions in the literature (for discrete- and continuous-time Markov chains) and looks at recent advances towards dynamically rich models such as probabilistic hybrid automata.

- Section 4 introduces notions of simulations, bisimulations and approximate bisimulations for classes of probabilistic models. For clarity, the presentation is divided into two parts: models with discrete spaces first and with continuous spaces later. Notice that these notions will be further investigated and will be the main topic of D.1.2, due on month 24.

2 Probabilistic and Hybrid Models

2.1 Classes of Markov Chains

This section introduces the basics from (discrete- and continuous-time) Markov chains and related extensions (Probabilistic Automata, Markov Decision Processes, Interactive Markov Chains). More extensive treatments on Markov chains can be found in many textbooks, such as [62, 63].

2.1.1 Discrete-time Markov chains

A discrete-time Markov chain (DTMC) has a possibly infinite yet countable set of states S which is equipped with an initial distribution μ_0; $\mu_0(s)$ for $s \in S$ denotes the probability that the DTMC will start in state s. The probability to move in one step from state s to a successor s' is given by a transition probability function $\mathbf{P}(s,s')$, for any pair $s,s' \in S$. We require each state to have at least one successor state (possibly the same state) that is reachable with a positive probability. Each state may satisfy certain atomic propositions, which is described by a mapping L. Labels that are associated to states are useful for the expression of time-dependent properties over the model, which is usually done via a modal logic such as PCTL [11]. Let AP denote the denumerable set of atomic propositions, i.e., the set of elementary properties, and let $\mathbb{B}$ the set of Booleans.

Definition 1 (DTMC)**.** *A* Discrete-Time Markov Chain $\mathscr{D}$ *is a tuple* $(S,\mathbf{P},L,\mu_0)$ *with a non-empty, countable set of* states S*,* transition probability function $\mathbf{P} : S \times S \to [0,1]$ *satisfying* $\sum_{s' \in S} \mathbf{P}(s,s') = 1$ *for all* $s \in S$*,* labeling function $L : S \times AP \to \mathbb{B}$*, and* initial distribution $\mu_0 : S \to [0,1]$.

A *path* in a DTMC $\mathscr{D} = (S,\mathbf{P},L,\mu_0)$ is an (infinite) sequence $\rho = s_0 s_1 s_2 \ldots$, sometimes written as $s_0 \to s_1 \to s_2 \ldots$. Let $\rho[i]$ denote the $(i{+}1)$-st state of ρ, i.e., $\rho[i] = s_i$. A finite prefix $\widehat{\rho} = s_0 s_1 s_2 \ldots s_l$ of a path $\rho = s_0 s_1 s_2 \ldots$ is called a path fragment. The last (first) element of $\widehat{\rho}$ is denoted by $last(\widehat{\rho})$ ($first(\widehat{\rho})$). We write $Path^{\mathscr{D}}(s)$ ($Path^{\mathscr{D}}_{fin}(s)$) for the set of all paths (path fragments, respectively) that start with state s.

The probability measure $Prob^{\mathscr{D}}$ on sets of paths is constructed as follows. Let Ω be the set $Path^{\mathscr{D}} = \bigcup_{s \in S} Path^{\mathscr{D}}(s)$, and $Cyl(s_0 \ldots s_k)$ be the (basic) cylinder set of $s_0 \ldots s_k$, i.e., the set of all paths in $Path^{\mathscr{D}}$ with prefix $s_0 \ldots s_k$. Let $\mathscr{B}$ be the σ-algebra generated by all cylinder sets over finite paths in DTMC $\mathscr{D}$. We define

$$Prob^{\mathscr{D}}(Cyl(s_0 \ldots s_k)) = \mu_0(s_0) \cdot \prod_{j=0}^{k-1} \mathbf{P}(s_j, s_{j+1})$$

Then, by Caratheodory's extension theorem [38], $Prob^{\mathscr{D}}$ extends to a probability measure on $\mathscr{B}$ in a unique manner. We sometimes write $Prob^{\mathscr{D}}_s$ instead of $Prob^{\mathscr{D}}$ in case s is the unique initial state, i.e., $\mu_0(s) = 1$ and $\mu_0(s') = 0$ for $s' \neq s$.

2.1.2 Probabilistic Automata

Probabilistic Automata (PA) constitute a mathematical framework for the specification and analysis of non-deterministic, probabilistic systems. They have been formally developed by Segala [85] to model and analyze asynchronous, concurrent systems with discrete probabilistic choice. PA are akin to Markov decision processes (MDP). A detailed comparison with models such as MDP, as well as generative and reactive probabilistic transition systems, is given in [83]. PAs are recognized as an adequate formalism for randomized distributed algorithms and fault tolerant systems, and are used as semantic models for formalisms such as probabilistic process algebras and probabilistic variant of Harel statecharts [60].

Definition 2 (Probabilistic automaton)**.** *A* probabilistic automaton $\mathscr{P}$ *is a collection* $(S, A, \rightarrow, \mu_0)$*, where*

- S *is a set of states,*
- A *is a set of actions,*
- $\rightarrow \subseteq S \times A \times Dist(S)$ *is a transition relation, and*
- $\mu_0 \in Dist(S)$ *is the initial probability distribution.*

Here $Dist(S)$ denotes the class of probability distribution functions over S. We will often express $(s, \alpha, \mu) \in \rightarrow$ as $s \xrightarrow{\alpha} \mu$. The intuitive operational behaviour of a PA is as follows. A PA starts in initial state s_0, possibly sampled from a distribution μ_0, selects one of the available (action, distribution) pairs that are enabled in s, i.e., an element from the set $\{(\alpha, \mu) \in A \times Dist(S) \mid s \xrightarrow{\alpha} \mu\}$. The PA then offers action α, which may possibly be subject to synchronisation with other PA, and moves to state s' with probability $\mu(s')$. Note that different pairs (α, μ) and (α, μ') may be enabled in s; however, for every action α and every state s there is at most one μ such that (α, μ) is enabled in s, in fact a Markov decision process (MDP) is obtained. If the set A contains a single element, the PA is semantically equivalent to a DTMC. Clearly, it follows that DTMC can also be directly seen as special instances of MDP, which is a perspective often used in the literature.

A *path* of a PA is an alternating, finite or infinite, sequence

$$s_0 \xrightarrow{\alpha_1,\mu_1} s_1 \xrightarrow{\alpha_2,\mu_2} s_2 \xrightarrow{\alpha_3,\mu_3} s_3 \ldots,$$

such that $s_i \xrightarrow{\alpha_{i+1}} \mu_{i+1}$ and $\mu_{i+1}(s_{i+1}) > 0$. The probability space for PA is defined analogously to that of DTMC.

2.1.3 Continuous-time Markov chains

While a DTMC is time-discrete, continuous-time Markov chains (CTMC) have an explicit reference to continuous time in the form of exit rates which determine, together with the transition probabilities, the stochastic evolution of the system in time.

Definition 3 (CTMC). *A* Continuous-Time Markov Chain $\mathscr{C}$ *is a tuple* $(S, \mathbf{P}, L, \mu_0, r)$ *with* S, $\mathbf{P}$, L, *and* μ_0 *as for DTMC, and additionally* exit rate function $r : S \to \mathbb{R}_{>0}$, *with* $\sup_{s \in S} r(s) \in \mathbb{R}_{>0}$.

The quantity $r(s)$ determines the random, exponentially distributed residence time in state s, that is, $1 - e^{-r(s)\cdot t}$ is the probability to take a transition emanating from s within the next t time units. If s is the current state, then a transition occurs after an average residence time of $\frac{1}{r(s)}$. The *time-dependent (one-step) transition probability* to move from s to s' within t time units is given by $\mathbf{P}(s, s', t) = \mathbf{P}(s, s') \cdot (1 - e^{-r(s)\cdot t})$. Note that self-loops are admitted, i.e., the successor state of s may be s.

A *path* in a CTMC $\mathscr{C} = (S, \mathbf{P}, L, \mu_0, r)$ is an alternating sequence $\sigma = s_0\, t_0\, s_1\, t_1\, s_2 \ldots$ with $s_i \in S$ and $t_i \in \mathbb{R}_{>0}$ for all $i \in \mathbb{N}$. The time stamps t_i denote the amount of time spent in state s_i. As the probability to reside zero time units in a state is zero, the t_i are strictly positive. A *path fragment* in a CTMC is a finite prefix of a path ending with a state. For a path $s_0\, t_0\, s_1\, t_1\, s_2 \ldots$ we may also write $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \ldots$. Again, $\sigma[i]$ denotes the $(i{+}1)$-st state of a path, i.e., $\sigma[i] = s_i$. By $\sigma@t$ we denote the state of σ occupied at time t, i.e. $\sigma@t = s_i$ where i is the smallest index such that $t < \sum_{j=0}^{i} t_j$. Similar to the case of DTMC, we write $\mathit{Path}^{\mathscr{C}}(s)$ ($\mathit{Path}^{\mathscr{C}}_{fin}(s)$) for the set of all paths (path fragments) of $\mathscr{C}$ that start with s, and, $\mathit{last}(\cdot)$ for the last state on a path fragment. By $\mathit{Path}^{\mathscr{C}}$, we denote the set of all paths in $\mathscr{C}$.

The probability measure $\mathit{Prob}^{\mathscr{C}}$ on the sample space $\Omega = \mathit{Path}^{\mathscr{C}}$ is defined in a similar way as for DTMC. A cylinder set $\mathit{Cyl}(s_0\, I_0\, s_1 \ldots I_{k-1}\, s_k)$, however, now depends on intervals $I_i = (0, z_i]$, $z_i \in \mathbb{R}_{>0}$, $i \in \{0, 1, \ldots, k{-}1\}$ and contains all paths $u_0\, t_0\, u_1\, t_1 \ldots \in \mathit{Path}^{\mathscr{C}}_{fin}$ with $u_i = s_i$, $t_i \in I_i$, for $0 \le i < k$, and $u_k = s_k$. Let $F(s, I)$ denote the probability of leaving state s within interval $I = (0, z]$, which is given by $F(s, I) = 1 - e^{-r(s)\cdot z}$. Then, $\mathit{Prob}^{\mathscr{C}}$ is given by

$$\mathit{Prob}^{\mathscr{C}}(\mathit{Cyl}(s_0\, I_0 \ldots I_{k-1}\, s_k)) = \mu_0(s_0) \cdot \prod_{j=0}^{k-1} \mathbf{P}(s_j, s_{j+1}) \cdot F(s_j, I_j)$$

and extends to a probability measure on $\mathscr{B}$ in a unique manner. As for the case of DTMC, we write $\mathit{Prob}^{\mathscr{C}}_s$ if s is the unique initial state. The *time-abstract* probabilistic behavior of CTMC $\mathscr{C} = (S, \mathbf{P}, L, \mu_0, r)$ is described by its *embedded DTMC* which is defined as $\mathit{emb}(\mathscr{C}) = (S, \mathbf{P}, L, \mu_0)$.

2.1.4 Interactive Markov chains

Interactive Markov chains (IMC) are models with semantics that distinguish between the advance of time and the occurrence of actions. This distinction leads to a behaviour where

two distinct phases are mixed. Phases during which one or more actions occur (together with their corresponding state changes), but where no time elapses, alternate with phases where time passes, but during which no actions happen. This separation of discrete and continuous phases is similar to that in timed automata [7]. This yields a mixture of labelled transition systems and CTMC, known as *interactive Markov chains* [57], in which action-labelled and rate-labelled transitions co-exist.

Definition 4 (Interactive Markov chain)**.** *An* interactive Markov chain *is characterized by a tuple* $\mathscr{I} = (S, A, \longrightarrow, \Longrightarrow, \mu_0)$ *where*

- S *is a nonempty set of states with* initial state $\mu_0 \in Dist(S)$*,*
- A *is a set of actions,*
- $\longrightarrow \subseteq S \times A \times S$ *is a set of* interactive *transitions, and*
- $\Longrightarrow \subseteq S \times \mathbb{R}_{>0} \times S$ *is a set of* Markovian *transitions.*

We abbreviate $(s, \alpha, s') \in \longrightarrow$ as $s \xrightarrow{\alpha} s'$ and similarly, $(s, \lambda, s') \in \Longrightarrow$ by $s \overset{\lambda}{\Longrightarrow} s'$. The interpretation of Markovian transition $s \overset{\lambda}{\Longrightarrow} s'$ is that the IMC can switch from state s to s' within d time units with probability $1 - e^{-\lambda \cdot d}$. The positive real value λ thus uniquely identifies a negative exponential distribution. For a state s, let $\mathbf{R}(s, s') = \sum\{\lambda \mid s \overset{\lambda}{\Longrightarrow} s'\}$ be the *rate* to move from state s to state s'. If $\mathbf{R}(s, s') > 0$ for more than one state s', a competition between the transitions of s exists, known as the race condition. The probability to move from such state s to a particular state s' within d time units, i.e., the Markovian transition $s \to s'$ wins the race, is given by:

$$\frac{\mathbf{R}(s, s')}{E(s)} \cdot \left(1 - e^{-E(s) \cdot d}\right),$$

where $E(s) = \sum_{s' \in S} \mathbf{R}(s, s')$ denotes the exit rate of state s. Intuitively, it states that after a delay of at most d time units (second term), the IMC moves probabilistically to a direct successor state s' with discrete branching probability $\mathbf{P}(s, s') = \frac{\mathbf{R}(s,s')}{E(s)}$.

Interactive transitions, also called τ-labeled interactive transitions, play a special role in IMC. They are not delayed. Thus, internal interactive transitions can be assumed to take place immediately. Now consider a state with both a Markovian transition with rate λ, say, and a τ-transition. Which transition can now occur? Since the τ-transition takes no time, it can be taken immediately. Instead, the probability that the Markovian transition executes immediately is zero. This justifies that interactive transitions take precedence over Markovian transitions. This is called the *maximal progress assumption*, which is also depicted in Figure 2.1.

Assumption 1 (Maximal progress assumption)**.** *In any IMC, interactive transitions take precedence over Markovian transitions.*

The probability space for IMC is defined analogously to that of CTMC.

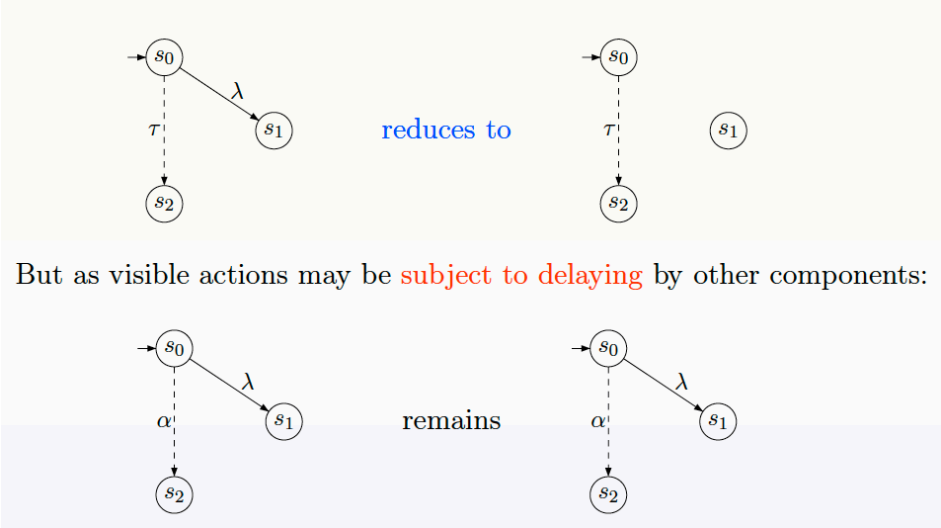


Figure 2.1: Maximal progress assumption in interactive Markov chains

2.2 Discrete-time Stochastic Hybrid Models

We define a discrete-time Stochastic Hybrid Model (DTSHS) as the discrete time counterpart of the general continuous time model described in [19], to be presented in Section 2.3.

The state of a DTSHS is characterized by two components: a discrete and a continuous one. The discrete state component takes on values in a countable set $\mathcal{Q}$ of modes, whereas the continuous state space in each mode $q \in \mathcal{Q}$ is given by the Euclidean space $\mathbb{R}^{n(q)}$, whose dimension $n(q)$ is mode-characteristic and determined by a map $n : \mathcal{Q} \to \mathbb{N}$. Thus the hybrid state space can be globally described as $\mathcal{S} := \cup_{q\in\mathcal{Q}}\{q\} \times \mathbb{R}^{n(q)}$, that is as the disjoint union of Euclidean domains associated to the respective discrete modes. Let us denote with $\mathcal{B}(\mathcal{S})$ the σ-field generated by the subsets of $\mathcal{S}$ of the form $\cup_{q\in\mathcal{Q}}\{q\} \times A_q$, where A_q is a Borel set in the space $\mathbb{R}^{n(q)}$. The hybrid space $\mathcal{S}$ is metrizable, namely it can be endowed with a metric that is equivalent to the usual Euclidean metric when restricted to each domain $\mathbb{R}^{n(q)}$ [25]. This shows that $(\mathcal{S}, \mathcal{B}(\mathcal{S}))$ is a Borel space with Polish structure, namely a space that is homeomorphic to a Borel subset of a complete separable metric space.

Qualitatively, the dynamics of the model can be described as follows. The continuous state of a DTSHS evolves according to a probabilistic law that depends on the actual operating mode. A discrete transition from the current operating mode to a different one may occur during the continuous state evolution, again according to some probabilistic law. This will in turn cause a modification of the probabilistic law governing the continuous state dynamics. A control input can affect the discrete and continuous evolution of the system. Moreover, after a discrete transition has occurred, the continuous state component is subject to a probabilistic reset that is also influenced by some control input. We distinguish this latter input from the former one, naming them respectively *reset* and *transition* input.

Definition 5 (DTSHS). *A discrete time stochastic hybrid system is a tuple*

$$\mathfrak{S} = (\mathscr{Q}, n, \mathscr{U}, \Sigma, T_x, T_q, T_r),$$

where

- $\mathscr{Q} := \{q_1, q_2, \ldots, q_m\}$*, for some* $m \in \mathbb{N}$*, represents the discrete state space,*
- $n : \mathscr{Q} \to \mathbb{N}$ *assigns to each discrete state value* $q \in \mathscr{Q}$ *the dimension of the continuous state space* $\mathbb{R}^{n(q)}$*,*
- $\mathscr{U}$ *is a compact Borel space representing the transition control space,*
- Σ *is a compact Borel space representing the reset control space,*
- $T_x : \mathscr{B}(\mathbb{R}^{n(\cdot)}) \times \mathscr{S} \times \mathscr{U} \to [0,1]$ *is a Borel-measurable stochastic kernel on* $\mathbb{R}^{n(\cdot)}$ *given* $\mathscr{S} \times \mathscr{U}$*, which assigns to each* $s = (q, x) \in \mathscr{S}$ *and* $u \in \mathscr{U}$ *a probability measure on the Borel space* $(\mathbb{R}^{n(q)}, \mathscr{B}(\mathbb{R}^{n(q)}))$*:* $T_x(\cdot|s,u)$*,*
- $T_q : \mathscr{Q} \times \mathscr{S} \times \mathscr{U} \to [0,1]$ *is a discrete stochastic kernel on* $\mathscr{Q}$ *given* $\mathscr{S} \times \mathscr{U}$*, which assigns to each* $s \in \mathscr{S}$ *and* $u \in \mathscr{U}$*, a probability distribution over* $\mathscr{Q}$*:* $T_q(\cdot|s,u)$*, and*
- $T_r : \mathscr{B}(\mathbb{R}^{n(\cdot)}) \times \mathscr{S} \times \Sigma \times \mathscr{Q} \to [0,1]$ *is a Borel-measurable stochastic kernel on* $\mathbb{R}^{n(\cdot)}$ *given* $\mathscr{S} \times \Sigma \times \mathscr{Q}$*, that assigns to each* $s \in \mathscr{S}$*,* $\sigma \in \Sigma$*, and* $q' \in \mathscr{Q}$*, a probability measure on the Borel space* $(\mathbb{R}^{n(q')}, \mathscr{B}(\mathbb{R}^{n(q')}))$*:* $T_r(\cdot|s,\sigma,q')$*.* □

To describe the semantics of a DTSHS, we need to specify an initial condition $s_0 \in \mathscr{S}$ (which may be sampled from an initial probability distribution) and how the reset and transition inputs are chosen. Here, we consider a DTSHS evolving over a finite time horizon $[0, N]$, with inputs chosen according to a Markov policy.

Definition 6 (Markov Policy). *A Markov policy* π *for a DTSHS* $\mathfrak{S} = (\mathscr{Q}, n, \mathscr{U}, \Sigma, T_x, T_q, T_r)$ *is a sequence* $\pi = (\pi_0, \pi_1, \pi_2, \ldots)$ *of measurable maps* $\pi_k : \mathscr{S} \to \mathscr{U} \times \Sigma$*,* $k = 0, 1, 2, \ldots$*, from the hybrid state space* $\mathscr{S} = \cup_{q \in \mathscr{Q}} \{q\} \times \mathbb{R}^{n(q)}$ *to the control input space* $\mathscr{U} \times \Sigma$*.* □

We denote the set of Markov policies as $\mathscr{M}_m$. With regards to the measurability of the function $\pi_k : \mathscr{S} \to \mathscr{U} \times \Sigma$, it is in general useful to assume universal measurability, as in [5]. This measurability condition is weaker than the Borel measurability condition and is needed to assess properties which hold uniformly in the initial condition s_0, particularly when policies are to be synthesized according to an optimization procedure [13]. For the scope of this report, Borel measurability is of course also sufficient.

The semantics of a DTSHS can be algorithmically defined through the notion of *execution*. In the sequel, we shall use boldface to denote random variables and normal typeset to denote sample values.

Definition 7 (Execution of a DTSHS). *Consider a DTSHS* $\mathfrak{S} = (\mathscr{Q}, n, \mathscr{U}, \Sigma, T_x, T_q, T_r)$*. A stochastic process* $\{\mathbf{s}(k) = (\mathbf{q}(k), \mathbf{x}(k)),\ k \geq 0\}$ *with values in* $\mathscr{S} = \cup_{q \in \mathscr{Q}} \{q\} \times \mathbb{R}^{n(q)}$ *is an execution of* $\mathfrak{S}$ *associated with a Markov policy* $\pi \in \mathscr{M}_m$ *and an initial condition* $s_0 = (q_0, x_0) \in \mathscr{S}$ *if its sample paths are obtained according to the following algorithm:*

set $\mathbf{q}(0) = q_0$, $\mathbf{x}(0) = x_0$, *and* $k = 0$;

for any $k > 0$ *do*

set $(u_k, \sigma_k) = \pi_k((q_k, x_k))$;

extract from $\mathscr{Q}$ *a value* q_{k+1} *for* $\mathbf{q}(k+1)$ *according to* $T_q(\cdot\,|(q_k, x_k), u_k)$;

if $q_{k+1} = q_k$, *then*

extract from $\mathbb{R}^{n(q_{k+1})}$ *a value* x_{k+1} *for* $\mathbf{x}(k+1)$ *according to* $T_x(\cdot\,|(q_k, x_k), u_k)$;

else

extract from $\mathbb{R}^{n(q_{k+1})}$ *a value* x_{k+1} *for* $\mathbf{x}(k+1)$ *according to* $T_r(\cdot\,|(q_k, x_k), \sigma_k, q_{k+1})$;

$k \to k+1$;

end. □

By appropriate choice of the discrete transition kernel T_q, it is possible to model the *spontaneous jumps* that *may* occur during the continuous state evolution, as well as the *forced jumps* that *must* occur when the continuous state exits some prescribed domain. As for the spontaneous transitions, if a discrete transition from q to $q' \neq q$ is enabled at $(q, x) \in \mathscr{S}$ by the control input $u \in \mathscr{U}$, then this can be encoded by the condition $T_q(q'|(q, x), u) > 0$. As for the forced transitions, the *invariant set* $Inv(q)$ associated with mode $q \in \mathscr{Q}$, namely the set of all the admissible values for the continuous state within q, can be expressed in terms of T_q by forcing $T_q(q|(q, x), u)$ to be equal to zero for all the continuous state values $x \in \mathbb{R}^{n(q)}$ outside $Inv(q)$, irrespectively of the value of the control input $u \in \mathscr{U}$. Thus $Inv(q) := \mathbb{R}^{n(q)} \setminus \{x \in \mathbb{R}^{n(q)} : T_q(q|(q, x), u) = 0, \forall u \in \mathscr{U}\}$, and as soon as $x \notin Inv(q)$ while the system evolves in mode q, a jump from q to some $q' \neq q$ is forced. Then, unlike the continuous time model in [19] (see Section 2.3), spatial guards here are implicitly defined through T_q.

In order to simplify notations and to relate the DTSHS model to general state-space Markov processes, let us introduce a stochastic kernel $\tau_x : \mathscr{B}(\mathbb{R}^{n(\cdot)}) \times \mathscr{S} \times \mathscr{U} \times \Sigma \times \mathscr{Q} \to [0, 1]$, defined on $\mathbb{R}^{n(\cdot)}$ given $\mathscr{S} \times \mathscr{U} \times \Sigma \times \mathscr{Q}$:

$$\tau_x(\cdot\,|(q, x), u, \sigma, q') = \begin{cases} T_x(\cdot|(q, x), u), & \text{if } q' = q \\ R(\cdot|(q, x), \sigma, q'), & \text{if } q' \neq q. \end{cases}$$

This kernel assigns to each $s = (q, x) \in \mathscr{S}$, $u \in \mathscr{U}$, $\sigma \in \Sigma$ and $q' \in \mathscr{Q}$ a probability measure on the Borel space $(\mathbb{R}^{n(q')}, \mathscr{B}(\mathbb{R}^{n(q')}))$. The kernel τ_x is used in the DTSHS algorithm to randomly select a value for the continuous state at time $k+1$, given the values taken by the hybrid state and the control input at time k, and that of the discrete state at time $k+1$.

Based on τ_x we can introduce the Borel-measurable stochastic kernel $T_s : \mathscr{B}(\mathscr{S}) \times \mathscr{S} \times \mathscr{U} \times \Sigma \to [0, 1]$ on $\mathscr{S}$ given $\mathscr{S} \times \mathscr{U} \times \Sigma$:

$$T_s((\cdot, q)\,|s, (u, \sigma)) = \tau_x(\cdot\,|s, u, \sigma, q) T_q(q|s, u),\ q \in \mathscr{Q}, \tag{2.1}$$

which assigns to each $s \in \mathscr{S}$, $(u, \sigma) \in \mathscr{U} \times \Sigma$ a probability measure on the Borel space $(\mathscr{S}, \mathscr{B}(\mathscr{S}))$.

Then, the DTSHS algorithm in Definition 7 can be rewritten in a more compact form as:

DTSHS algorithm

set $\mathbf{s}(0) = s_0$ *and* $k = 0$*;*

for any $k > 0$ *do*

set $(u_k, \sigma_k) = \pi_k(s_k)$*;*

extract from $\mathscr{S}$ *a value* s_{k+1} *for* $\mathbf{s}(k+1)$ *according to* $T_s(\cdot\,|s_k, (u_k, \sigma_k))$*;*

$k \to k+1$;

end. □

This shows that a DTSHS $\mathfrak{S} = (\mathscr{Q}, n, \mathscr{U}, \Sigma, T_x, T_q, T_r)$ can be described as a controlled Markov process with state space $\mathscr{S} = \cup_{q\in\mathscr{Q}}\{q\} \times \mathbb{R}^{n(q)}$, control space $\mathscr{A} := \mathscr{U} \times \Sigma$, and controlled transition probability function $T_s : \mathscr{B}(\mathscr{S}) \times \mathscr{S} \times \mathscr{A} \to [0, 1]$ defined in (2.1) [81].

As a consequence of this representation of $\mathfrak{S}$, the execution $\{\mathbf{s}(k) = (\mathbf{q}(k), \mathbf{x}(k)),\ k \geq 0\}$ associated with $s_0 \in \mathscr{S}$ and $\pi \in \mathscr{M}_m$ is a stochastic process defined on the canonical sample space $\Omega = \mathscr{S}^\infty$, endowed with its product topology $\mathscr{B}(\Omega)$, with probability measure $Prob_{s_0,\pi}$ uniquely defined by the transition kernel T_s, the policy $\pi \in \mathscr{M}_m$, and the initial condition $s_0 \in \mathscr{S}$ [13, Proposition 7.45]. It also follows that the execution of a DTSHS associated with a Markov policy $\pi = (\pi_0, \pi_1, \dots) \in \mathscr{M}_m$ and an initial condition s_0 is an inhomogeneous Markov process with one-step transition kernels $T_s(\cdot|s, \pi_k(s))$, $k \geq 0$.

2.3 Continuous-time Stochastic Hybrid Models

2.3.1 General Stochastic Hybrid System model

In this section, we describe and elaborate the general stochastic hybrid system (GSHS) model that was first introduced in [19], and later refined in [18]. This model is a continuous-time version of that presented in Section 2.2.

In a GSHS, the continuous evolution of the state is confined to some domain (to be precisely defined shortly) and characterized by a stochastic differential equation (SDE, [76]) with drift and diffusion terms that are state-depent. As for the discrete state evolution, both spontaneous transitions (transitions that occur according to probabilistic arrivals) and forced transitions (transitions that are driven by a boundary hitting time) are allowed. As a result of a discrete transition, the hybrid state is subject to a reset according to a probabilistic map.

In the definition below, we focus on a GSHS where the discrete state reset does not admit non-determinism (this raises a semantical difference with the PHA in Section 2.4). For the sake of clarity, we distinguish between the probabilistic reset maps governing spontaneous

and forced transitions. The two reset maps can in fact shown to be equivalent to a single reset map, as in the original GSHS model [18].

Definition 8 (General Stochastic Hybrid System)**.** *A* General Stochastic Hybrid System *is a collection*

$$\mathfrak{S} = (\mathcal{Q}, n, A, B, \Gamma, R^{\Gamma}, \Lambda, R^{\Lambda}, \pi),$$

where

- $\mathcal{Q} = \{q_1, q_2, \ldots, q_m\}, m \in \mathbb{N}$ *is a countable set of discrete modes representing the discrete state space,*
- $n : \mathcal{Q} \to \mathbb{N}$ *is a map that determines the dimension of the continuous state space associated with each mode. For* $q \in \mathcal{Q}$*, the continuous state space is the Euclidean space* $\mathbb{R}^{n(q)}$*. The hybrid state space is then* $\mathcal{S} = \cup_{q\in\mathcal{Q}}\{q\} \times \mathbb{R}^{n(q)}$*,*
- $A = \{a(q,\cdot) : \mathbb{R}^{n(q)} \to \mathbb{R}^{n(q)}, q \in \mathcal{Q}\}$ *is the collection of drift terms of the SDEs governing the continuous dynamics,*
- $B = \{b(q,\cdot) : \mathbb{R}^{n(q)} \to \mathbb{R}^{n(q)\times n(q)}, q \in \mathcal{Q}\}$ *is the collection of diffusion terms of the SDEs governing the continuous dynamics,*
- Γ *is a subset of the hybrid state space* $\mathcal{S}$ *defined as* $\Gamma = \cup_{q\in\mathcal{Q}}\{q\} \times \Gamma_q$*, where* $\Gamma_q = \cup_{q'\neq q\in\mathcal{Q}}\gamma_{qq'}$ *is a closed set composed of* $m-1$ *disjoint guard sets* $\gamma_{qq'}$ *causing forced transitions from mode* q *to mode* $q' \neq q$*,*
- $R^{\Gamma} : \mathcal{B}(\mathbb{R}^{n(\cdot)}) \times \mathcal{Q} \times \Gamma \to [0,1]$ *is the reset stochastic kernel associated with* Γ*. Specifically,* $R^{\Gamma}(\cdot|q',(q,x))$ *is a probability measure defined on* $\mathbb{R}^{n(q')} \setminus \Gamma_{q'}$*, which describes the probabilistic reset of the continuous state when a jump from mode* q *to* q' *occurs from* $x \in \gamma_{qq'}$*,*
- $\Lambda : \mathcal{S} \setminus \Gamma \times \mathcal{Q} \to \mathbb{R}^{+}$ *is the transition intensity function governing spontaneous transitions. Specifically, for any* $q \neq q' \in \mathcal{Q}$*,* $\lambda_{qq'}(x) := \Lambda((q,x),q')$ *is the jump rate from mode* q *to mode* q' *when* $x \in \mathbb{R}^{n(q)} \setminus \Gamma_q$*,*
- $R^{\Lambda} : \mathcal{B}(\mathbb{R}^{n(\cdot)}) \times \mathcal{Q} \times \mathcal{S} \setminus \Gamma \to [0,1]$ *is the reset stochastic kernel associated with* Λ*. In particular,* $R^{\Lambda}(\cdot|q',(q,x))$ *is a probability measure defined on* $\mathbb{R}^{n(q')} \setminus \Gamma_{q'}$ *that describes the probabilistic reset of the continuous state when a jump from mode* q *to* q' *occurs from* $x \in \mathbb{R}^{n(q)} \setminus \Gamma_q$*, and*
- $\pi : \mathcal{B}(\mathcal{S}) \to [0,1]$ *is a probability measure concentrated on* $\mathcal{S} \setminus \Gamma$ *that describes the initial state distribution.* □

As in the case of DTSHS, in the definition above $\mathcal{B}(\mathcal{S})$ denotes the σ-field generated by the subsets of $\mathcal{S}$ of the form $C = \cup_q\{q\} \times C_q$, where C_q is a Borel set in $\mathbb{R}^{n(q)}$. $\mathcal{S}$ is metrizable and admits a topology as above.

Note that the evolution of the continuous state within each mode $q \in \mathcal{Q}$ is confined to the open set $X_q := \mathbb{R}^{n(q)} \setminus \Gamma_q$, which matches the GSHS definition in [18], where an open domain $X_q \subseteq \mathbb{R}^{n(q)}$ is associated to each mode $q \in \mathcal{Q}$. This is because possible resets from within $\mathcal{S} \setminus \Gamma$ maintain the hybrid state within $\mathcal{S} \setminus \Gamma$ (spontaneous transitions), and, as soon as the hybrid state reaches its boundary, it is reset within $\mathcal{S} \setminus \Gamma$ (forced transitions).

To define the semantics of a GSHS through the notion of *execution*, we first need to introduce some assumptions and clarify some notations.

Assumption 2 (On the dynamics within each mode)**.** *The drift and diffusion terms $a(q,\cdot)$ and $b(q,\cdot)$ are bounded and uniformly Lipschitz continuous for any $q \in \mathcal{Q}$.* □

This assumption guarantees the existence and uniqueness of the solution to the SDE associated with $q \in \mathcal{Q}$, namely

$$d\mathbf{v}(t) = a(q,\mathbf{v}(t))dt + b(q,\mathbf{v}(t))d\mathbf{w}_q(t), \tag{2.2}$$

where $\mathbf{w}_q$ is a $n(q)$ dimensional standard Wiener process, for an arbitrarily fixed initial condition $\mathbf{v}(0)$, independent of $\mathbf{w}_q$. The solution process $\mathbf{v}(t)$, $t \geq 0$, is Markov and continuous.

Let $\lambda : \mathcal{S} \setminus \Gamma \to \mathbb{R}^+$ be the jump rate function associating to the hybrid state $s = (q,x) \in \mathcal{S} \setminus \Gamma$ the jump rate

$$\lambda(q,x) = \sum_{q' \in \mathcal{Q}, q' \neq q} \lambda_{qq'}(x).$$

Assumption 3 (On the spontaneous transitions)**.** *The jump rate function $\lambda : \mathcal{S} \setminus \Gamma \to \mathbb{R}^+$ satisfies the following conditions:*

1. *it is measurable and bounded, and*
2. *for any $q \in \mathcal{Q}$, and any sample path $\omega^{(q,x)}(t)$, $t \geq 0$, of the diffusion process solving equation (2.2) initialized at $x \in \mathbb{R}^{n(q)} \setminus \Gamma_q$, there exists $\epsilon_q(x) > 0$ such that $\lambda(q, \omega^{(q,x)}(t))$ is integrable over $[0, \epsilon_q(x))$.* □

By Assumption 3, the maximum jump rate from mode $q \in \mathcal{Q}$ is bounded:

$$\hat{\lambda}_q = \sup_{x \in \mathbb{R}^{n(q)} \setminus \Gamma_q} \lambda(q,x) < \infty,$$

which ensures that no multiple instantaneous jumps are allowed.

Define the hybrid reset kernel $R : \mathcal{B}(\mathcal{S}) \times \mathcal{S} \to [0,1]$ governing both the forced and spontaneous resets of the hybrid state from $(q,x) \in \mathcal{S}$ as

$$R(C|(q,x)) = \begin{cases} \displaystyle\sum_{q' \neq q \in \mathcal{Q}} R^{\Lambda}(C_{q'}|q',(q,x)) \frac{\lambda_{qq'}(x)}{\lambda(q,x)}, & x \in \mathbb{R}^{n(q)} \setminus \Gamma_q, \\ R^{\Gamma}(C_{q'}|q',(q,x)), & x \in \gamma_{qq'},\ q' \neq q \in \mathcal{Q}, \end{cases} \tag{2.3}$$

where $C \in \mathcal{B}(\mathcal{S})$ and $C_{q'} = \{x \in \mathbb{R}^{n(q')} : (q',x) \in C\}$, $q' \in \mathcal{Q}$.

Since both continuous reset kernels $R^{\Gamma}(\cdot|q',(q,x))$ and $R^{\Lambda}(\cdot|q',(q,x))$ are probability measures concentrated on $\mathbb{R}^{n(q')} \setminus \Gamma_{q'}$, for any $(q,x) \in \mathcal{S}$ the hybrid reset kernel $R(\cdot|(q,x))$ is a probability measure concentrated on $\mathcal{S} \setminus \Gamma$. We also require the following.

Assumption 4 (On the hybrid state resets)**.** *For all $C \in \mathscr{B}(\mathscr{S})$, $R(C|\cdot) : \mathscr{S} \to [0,1]$ is measurable.* □

Thus, $R : \mathscr{B}(\mathscr{S}) \times \mathscr{S} \to [0,1]$ is a transition measure. Extracting a value from $R(\cdot|s)$ is equivalent to first extracting a value for a random variable uniformly distributed in the Hilbert hypercube and then using an appropriate map derived from $R(\cdot|s)$, [25].

Let $\omega^{(q,x)}(t)$, $t \geq 0$, be a continuous path taking values in $\mathbb{R}^{n(q)}$ starting from $\omega^{(q,x)}(0) = x$. Define

$$F(t, \omega^{(q,x)}) = I_{\{t<t_\star(\omega^{(q,x)})\}} e^{-\int_0^t \lambda(q,\omega^{(q,x)}(\tau))d\tau} \tag{2.4}$$

as the *survivor function* at time t of the path $\omega^{(q,x)}(\cdot)$, where $t_\star(\omega^{(q,x)})$ is the time instant when $\omega^{(q,x)}$ first hits the guard set Γ_q. Within the definition of a GSHS execution, the survivor function $F(t, \omega^{(q,x)})$ describes the probability that no transition (neither forced nor spontaneous) happens along a time horizon of length t, while the continuous state is evolving according to path $\omega^{(q,x)}$ within mode q starting from $x \in \mathbb{R}^{n(q)}$. The indicator function $I_{\{t<t_\star(\omega^{(q,x)})\}}$ accounts for forced transitions, whereas the exponential $e^{-\int_0^t \lambda(q,\omega^{(q,x)}(\tau))d\tau}$ accounts for spontaneous transitions.

We are now in a position to semantically define a GSHS through the notion of execution.

Definition 9 (Execution)**.** *Consider a GSHS $\mathfrak{S} = (\mathscr{Q}, n, A, B, \Gamma, R^\Gamma, \Lambda, R^\Lambda, \pi)$. A stochastic process $\{\mathbf{s}(t) = (\mathbf{q}(t), \mathbf{x}(t)),\ t \geq 0\}$ with values in $\mathscr{S} = \cup_{q\in\mathscr{Q}}\{q\} \times \mathbb{R}^{n(q)}$ is an execution of $\mathfrak{S}$ if its sample paths $s(t)$, $t \geq 0$, are obtained according to the following algorithm:*

extract at random from π a value $s(0) = (q(0), x(0))$ for $\mathbf{s}(0)$;

set $\hat{q} = q(0)$, $\hat{x} = x(0)$, $k = 1$ and $\tau = 0$;

while $\tau < \infty$ do

  extract a sample path $\omega^{(\hat{q},\hat{x})}(t)$, $t \geq 0$ of the diffusion process solving the SDE (2.2) with $q = \hat{q}$ initialized with $\mathbf{v}(0) = \hat{x}$;

  compute the jump time instant $\hat{T}_k = \tau + \inf\{t > 0 : F(t, \omega^{(\hat{q},\hat{x})}(t)) \leq e^{-\hat{\lambda}_{\hat{q}} t}\}$;

  if $\hat{T}_k = \infty$ then

    set $s(t) = (\hat{q}, \omega^{(\hat{q},\hat{x})}(t-\tau))$, $t \geq \tau$;

  else

    set $s(t) = (\hat{q}, \omega^{(\hat{q},\hat{x})}(t-\tau))$, $t \in [\tau, \tau + \hat{T}_k)$;

    extract a value $s(\tau + \hat{T}_k)$ for $\mathbf{s}(\tau + \hat{T}_k)$ according to $R(\cdot|(\hat{q}, \omega^{(\hat{q},\hat{x})}(\tau + \hat{T}_k)))$;

    set $(\hat{q}, \hat{x}) = s(\tau + \hat{T}_k)$;

  end

  set $\tau = \hat{T}_k$;

  increment k by one $k \to k + 1$;

end

In the algorithm, the Wiener processes, the initial state $\mathbf{s}(0)$*, and the extractions from the reset kernel* $R(\cdot|(q,x))$ *are independent.* □

Note that a sample-path of a GSHS execution is a concatenation of the sample paths of diffusion processes generated at each stopping time instant when a discrete transition occurs (Markov string process, [18]). Each single diffusion process is the solution to the SDE (2.2) initialized with the value of the continuous state updated through the hybrid reset kernel. The resulting GSHS sample-paths are right-continuous $\mathcal{S}$-valued functions on $[0,\infty)$, with left-limits on $(0,\infty)$ (*càdlàg*). The stochastic process in Definition 9 is constructed on the set $\Omega = D_{\mathcal{S}}[0,\infty)$ of all $\mathcal{S}$-valued càdlàg functions. The topology induced on $D_{\mathcal{S}}[0,\infty)$ by the so-called *Skorokhod metric* allows to study convergence in distribution of stochastic processes with jumps, and on the space of continuous $\mathcal{S}$-valued functions on $[0,\infty)$ coincides with the topology of uniform convergence, [39].

Assumption 5. *For any execution associated with* $\pi = \delta_s$*,* $s \in \mathcal{S} \setminus \Gamma$ *(namely, if the initial distribution is concentrated on point s), the expected value of the number of jumps within the time interval* $[0,t]$ *is bounded for all* $t \geq 0$*:* $\mathbb{E}_s[N_t] < \infty$*,* $t \geq 0$*, where* $N_t = \sum_{k\geq 1} I(t \geq \hat{T}_k)$*.* □

Assumption 5 is used to exclude from the GSHS execution Zeno realizations where an infinite number of discrete transitions are taken within a finite time. In the statement, the symbol $\mathbb{E}_s$ denotes a conditional expectation related to the random variables $\hat{T}_k$.

Proposition 1 (GSHS execution, [18])**.** *Consider a GSHS* $\mathfrak{S}$*. Under the previous assumptions, the execution* $\mathbf{s}(t), t \geq 0$*, of* $\mathfrak{S}$ *is a càdlàg strong Markov process.* □

In particular, in [18] it is shown that the càdlàg and Markov properties of the component diffusion processes are preserved in the hybrid case, by constructing *strings* of such processes. For other contributions along this line of work, please refer to [25] and [47].

From a different perspective, when the continuous state space dimension is not mode-dependent, a GSHS can be described through a stochastic differential equation on a hybrid state space by using Poisson random measures, [46] and [65]. Within this approach, [15] proves that the Markov property of the concatenated diffusion processes originating a switching diffusion with probabilistic hybrid jumps due to spontaneous transitions are preserved under assumptions that are equivalent to those in Proposition 1.

By Proposition 1, a GSHS generates $(\Omega, \mathscr{F}, (\mathscr{F}_t)_{t\geq 0}, (\mathbf{s}(t))_{t\geq 0}, \mathscr{P}_s)$, where $s \in \mathcal{S} \setminus \Gamma$, a family of strong Markov processes with state space $(\mathcal{S}, \mathscr{B}(\mathcal{S}))$. Here $(\Omega, \mathscr{F}, (\mathscr{F}_t)_{t\geq 0})$ is the canonical space with $\Omega = D_{\mathcal{S}}[0,\infty)$, $(\mathscr{F}_t)_{t\geq 0}$ is the natural filtration of the coordinate function (i.e. $\mathscr{F}_t = \sigma(\omega(s), s \leq t)$, $\omega \in D_{\mathcal{S}}[0,\infty)$) and $\mathscr{F}$ is the smallest σ-field containing all $(\mathscr{F}_t)_{t\geq 0}$. $(\mathbf{s}(t))_{t\geq 0}$ is a family of random variables such that $\mathbf{s}(t)$ is $\mathscr{F}_t$-measurable for all $t \geq 0$, and $\mathscr{P}_s$ is a probability measure on $(\Omega, \mathscr{F})$ such that $\mathbf{s}(t)$, $t \geq 0$, is a strong Markov process on $(\Omega, \mathscr{F}, \mathscr{P}_s)$ with transition function

$$p_t(s, C) = \mathscr{P}_s(\mathbf{s}(t) \in C),\ C \in \mathscr{B}(\mathcal{S}), t \geq 0,$$

and initial distribution δ_s, i.e., $\mathscr{P}_s(\mathbf{s}(0) = s) = 1$. $\mathbb{E}_s$ is the expectation defined via the probability measure $\mathscr{P}_s$. Notice the relationship between the probability measure $\mathscr{P}_s$ here introduced, and $Prob_s$ defined for the earlier models.

GSHS Generator. Let $(\Omega, \mathscr{F}, (\mathscr{F}_t)_{t\geq 0}, (\mathbf{s}(t))_{t\geq 0}, \mathscr{P}_s)$ be the family of strong Markov processes with state space $(\mathscr{S}, \mathscr{B}(\mathscr{S}))$. Consider the space $\mathscr{B}_b(\mathscr{S})$ of real-valued, bounded and measurable functions f on $(\mathscr{S}, \mathscr{B}(\mathscr{S}))$. For any $t \in \mathbb{R}^+$, define the *operator* $P_t : \mathscr{B}_b(\mathscr{S}) \to \mathscr{B}_b(\mathscr{S})$:

$$P_t f(s) = \mathbb{E}_s[f(\mathbf{s}(t))].$$

Due to the Markov property, P_t is a semigroup. It is possible to associate to P_t its *strong generator* $\mathscr{L}$ (also names *infinitesimal* generator), which can be thought of as the derivative of P_t at $t = 0$. Let $\mathscr{D}(\mathscr{L}) \subseteq \mathscr{B}_b(\mathscr{S})$ be the set of functions f for which the following limit exists

$$\lim_{t\to 0} \frac{1}{t}(P_t f - f),$$

where convergence is in the sup norm. If we denote the above limit as $\mathscr{L}f$, then

$$\lim_{t\to 0} \sup_{s\in\mathscr{S}} \left| \frac{1}{t}(P_t f(s) - f(s)) - \mathscr{L}f(s) \right| = 0.$$

The strong generator $\mathscr{L}$ is then defined by specifying the limit $\mathscr{L}f$, as well as its domain $\mathscr{D}(\mathscr{L})$. The following holds [25, Proposition 14.13]:

Proposition 2. *For $f \in \mathscr{D}(\mathscr{L})$, define the real-valued process*

$$M(t) = f(\mathbf{s}(t)) - f(\mathbf{s}(0)) - \int_0^t \mathscr{L}f(\mathbf{s}(u))du.$$

For any initial condition $\mathbf{s}(0) = s \in \mathscr{S}$, $M(t)$, $t \geq 0$, is a martingale on $(\Omega, \mathscr{F}, (\mathscr{F}_t)_{t\in\mathbb{R}^+}, \mathscr{P}_s)$. □

The knowledge of a strong generator allows one to characterize its associated stochastic process. It is also possible to prove convergence for Markov processes based on the limiting behavior of their strong generator, [39]. This should suggest that it is desirable to find an explicit form for the generator of a stochastic process that is an execution of a stochastic model. We are furthermore interested in introducing a weaker notion of generator.

Definition 10 (Extended Generator)**.** *Let $\mathscr{D}(\mathscr{L}_g) \subseteq \mathscr{B}_b(\mathscr{S})$ denote the set of functions $f : \mathscr{S} \to \mathbb{R}$ for which there exists a function $h : \mathscr{S} \to \mathbb{R}$, $h \in \mathscr{B}_b(\mathscr{S})$ such that the real-valued process*

$$M(t) = f(\mathbf{s}(t)) - f(\mathbf{s}(0)) - \int_0^t h(\mathbf{s}(u))du \tag{2.5}$$

is a local martingale on $(\Omega, \mathscr{F}, (\mathscr{F}_t)_{t\in\mathbb{R}^+}, \mathscr{P}_s)$ for any initial condition $\mathbf{s}(0) = s \in \mathscr{S}$. Then, the operator $\mathscr{L}_g : \mathscr{D}(\mathscr{L}_g) \to \mathscr{B}_b(\mathscr{S})$ defined by $h = \mathscr{L}_g f$ is the extended generator *of $(\Omega, \mathscr{F}, (\mathscr{F}_t)_{t\geq 0}, (\mathbf{s}(t))_{t\geq 0}, \mathscr{P}_s)$, $s \in \mathscr{S}$, with domain $\mathscr{D}(\mathscr{L}_g)$.* □

The above definition is justified in terms of Proposition 2, because the concept of local martingale is weaker and easier to characterize than that of martingale [14]. We say that a càdlàg Markov process $\mathbf{s}(t)$, $t \geq 0$, with values in $\mathscr{S}$, is a *solution* of the martingale problem $(\mathscr{L}, s_0)$ [local martingale problem (g, s_0)], if for any $f \in \mathscr{D}(\mathscr{L})$ [$\mathscr{D}(\mathscr{L}_g)$], $M(t)$ in Proposition 2 [equation (2.5)] is a martingale [a local martingale].

Consider a real-valued function f of the hybrid state space $\mathscr{S}, f : \mathscr{S} \to \mathbb{R}$.

Assumption 6. *Assume f is a class $C_b^2(\mathscr{S})$ function.* □

Here $C_b^2(\mathscr{S})$ is the class of real-valued, twice continuously differentiable and bounded functions on $\mathscr{S}$. The extended generator for the process associated with the GSHS model in Definition 8 is derived along the ideas developed in the seminal work of [25], and in the extension to the diffusion case presented in [19].

Let $\frac{\partial f(q,x)}{\partial x} a(q,x) = \sum_{i=1}^{n(q)} \frac{\partial f(q,x)}{\partial x_i} a_i(q,x)$ be the Lie derivative of $f(q,\cdot)$ along $a(q,\cdot)$, and $H_f(q,x) = \left[\frac{\partial^2 f(q,x)}{\partial x_i \partial x_j}\right]_{i,j=1,2,\ldots,n(q)}$ be the Hessian of $f(q,\cdot)$.

Proposition 3 (Extended Generator of $\mathfrak{S}$)**.** *The extended generator $\mathscr{L}_g : \mathscr{D}(\mathscr{L}_g) \to \mathscr{B}_b(\mathscr{S})$ associated with the executions of $\mathscr{S}_g$ is given by*

$$\mathscr{L}_g f(s) = \mathscr{L}_g^d f(s) + I_{\mathscr{S}\setminus\Gamma}(s) \sum_{q' \in \mathscr{Q}, q' \neq q} \lambda_{qq'}(x) \int_{\mathbb{R}^{n(q')}} \big(f((q',z)) - f(s)\big) R^{\Lambda}(dz|q',s),$$

$s = (q,x) \in \mathscr{S} \setminus \Gamma$, where

$$\mathscr{L}_g^d f(s) = \sum_{q \in \mathscr{Q}} \frac{\partial f(q,x)}{\partial x} a(q,x) + \frac{1}{2} Tr\big(b(q,x) b(q,x)^T H_f(q,x)\big).$$

The domain $\mathscr{D}(\mathscr{L}_g)$ of $\mathscr{L}_g$ is the set of functions $f \in C_b^2(\mathscr{S})$ satisfying the condition:

$$f(s) = \sum_{q' \in \mathscr{Q}, q' \neq q} \int_{\mathbb{R}^{n(q')}} f((q',z)) R^{\Gamma}(dz|q',s), \;\; s \in \Gamma.$$

□

Remark 1. *Proposition 3 follows from [19, Theorem 2], which in turn strictly adheres to [25, 26.14]. Notice that, unlike those sources, no bounded-variation of the expected value of the functions f is assumed. This is thanks to the simplifying boundedness hypothesis for f in Assumption 6, and on Assumption 5 for the GSHS model, which excludes Zeno behaviors (that is, an infinite number of transitions in any finite time interval). In the expression of $\mathscr{L}_g$, the first term $\mathscr{L}_g^d$ represents the contribution of the continuous part, as it encompasses operations on its drift and diffusion terms. The second describes the influence of the spontaneous transitions. These terms act on points of the hybrid state space that do not belong to the guard set Γ. Finally, the condition at the bottom line accounts for the resets due to the spatial constraints, and in fact acts on hybrid points belonging to Γ.* □

2.3.2 General Stochastic Hybrid Systems: Special Cases

GSHS include as special cases other classes of stochastic hybrid systems, such as PDMP [25](where the continuous state evolution within each mode is deterministic), SDP [47](where only spontaneous transitions can occur but with no reset of the continuous state), SDP with random hybrid jumps [15, 46](where only spontaneous transitions can occur causing a probabilistic reset of the whole hybrid state), and SDP with random and deterministic jumps [46](where the switching diffusion model with random hybrid jumps is enhanced with deterministic resets). We present explicitly a few connections in the following.

If the diffusion term B in $\mathfrak{S}$ is neglected, the extended generator $\mathscr{L}_g$ is included in that of PDMP, as formally derived in [25, 26.14]. More precisely, their structures coincide, but PDMP would have a larger domain of definition (that of functions of class $C_b^1(\mathscr{S})$).

A GHSH $\mathfrak{S}$ with the same continuous state space in each mode, no spatial guards ($\Gamma = \emptyset$), and (deterministic) identity reset maps ($\forall q, q' \in \mathscr{Q}, q' \neq q, R^{\Lambda}((q', \{x\})|(q, x)) = 1$) is a SDP. The extended generator for this special case coincides with that derived in the literature, for instance in [47], modulo the neglection of the two reset conditions, as well as the condition involving Γ.

Furthermore, in the purely deterministic (no diffusion terms, no transition intensities, nor discrete events and corresponding reset kernels) and single-domain case, the extended generator coincides with the Lie derivative of the function f taken along the corresponding vector field, [82].

2.4 Probabilistic Hybrid Automata

Probabilistic hybrid automata (PHA) differ from the aforementioned hybrid models in offering two distinct forms of branching behavior, namely non-deterministic and probabilistic branching. Historically, PHA have been derived from hybrid automata rather than Markov models. PHA extend the concept of hybrid automata with probabilistic transitions modeling, e.g., component failures or collisions on a shared medium and the resulting loss of data packets in distributed systems. Like dense-time hybrid automata, PHA feature a finite set of discrete locations or modes, each of which comes decorated with a (possibly non-linear) differential equation governing the dynamics of a vector of continuous variables while residing in the mode and with an invariant constraining the evolution of the continuous variables while in the mode. Mode changes are effected by instantaneous transitions guarded by conditions on the current values of the continuous variables, and such mode changes may also involve discontinuous updates to the continuous variables. Both transition selection and variable updates may be non-deterministic; the first situation arises in case of overlapping guard conditions, the second due to underspecification in the pre-post-relations defining these assignments. Beyond these mechanisms from hybrid automata, PHA add the probabilistic selection of a transition variant based on a random experiment: Following the idea of Sproston [86, 87], potentially non-deterministic transition selection happens first, choosing one transition among the (inherently finitely many) enabled ones, and is then followed by randomized selection of a transition variant (again from a finite set of variants) according to a probability distribution belonging to the selected transition. The different transition

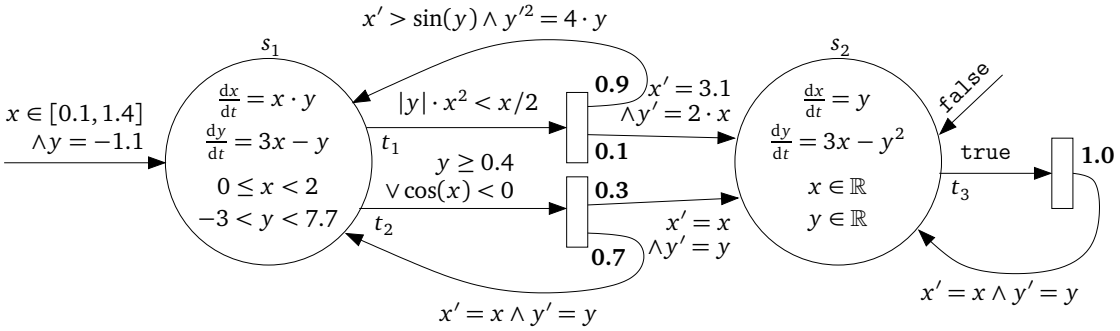


Figure 2.2: A probabilistic hybrid automaton A. The rectangular boxes denote random events selecting a transition variant with the probabilities denoted along the transition arcs. Note that transition selection and execution is a three-stage process: Non-determinism, which may arise due to overlapping guard conditions, is resolved strictly before performing the random experiment selecting a transition variant determining the successor location. Hereafter, a continuous successor state is selected.

variants can lead to different follow-up locations and different continuous successors, as depicted in Figure 2.2, where the guard condition determining transition selection is depicted to the left of the box denoting the random experiment. In comparison to Piecewise Deterministic Markov Processes (PDMP) [26] as well as GSHS (Sect. 2.3), PHA add a second form of choice dynamics, namely non-deterministic choice in both the selection among competing transitions and the computation of the continuous successor state. They are, however, more confined in their continuous dynamics: PHA do neither provide the transition rates of PDMPs nor the SDE present in GSHS.

The rationale for adding non-determinism on top of stochasticity is enhanced expressiveness when addressing open dynamic systems or partially developed technical systems. For open dynamic systems, the $1\frac{1}{2}$-player game semantics offered by a combination of non-determinism and stochasticity permits an encoding of control problems, where the deterministic and the stochastic dynamics represent the dynamics of the controlled system, while non-deterministic branching encodes the options available to the controller, i.e. the controllable actions. For partially developed technical systems, where only part of the uncertain system dynamics, like component failures in the subsystem modelled, can be characterized statistically, some dynamical aspects are necessarily unknown at the time of modeling and analysis and thus need to be accounted for in their worst-case form, as permitted non-deterministic underspecification under a demonic interpretation of non-determinism.

Upon a run, the probabilistic hybrid automaton engages in a sequence of continuous flows and discrete jumps, where the continuous flows are solutions to the ordinary differential equations assigned to the current location and the discrete jumps coincide with enabled transitions, thereby first selecting non-deterministically among the enabled transitions and then probabilistically among the different target locations and finally again non-deterministically among the assignments to continuous variables permitted by the transi-

tion. As usual in models blending probabilistic and non-deterministic choice, like Markov Decision Processes [12], we assume that the dynamics is controlled by a *decision maker* or *policy (scheduler, adversary)* resolving the non-determinism based on (complete) observation of the current state and history. Typical analysis goals are to construct a policy achieving a given system requirement (control synthesis) or to prove that a system requirement holds irrespective of the particular policy chosen (verification of open systems). Thereby, requirements typically take the form of probabilistic avoid properties, demanding that the probability of avoiding undesirable states remains within a given tolerance.

Formally, a (flat, we will introduce concurrency later on) *dense-time probabilistic hybrid automaton* $A = (\Lambda, \text{Trans}, R, s, p, g, \text{asgn}, \text{ode}, \text{inv}, \text{init})$ consists of the following components:

- Finite sets Λ of *locations*, Trans of *transitions*, and $R = \{x_1, \ldots, x_n\}$ of *continuous state components*, together with (total) mappings $s : \text{Trans} \xrightarrow{\text{total}} \Lambda$, assigning to each transition its source location, and $p : \text{Trans} \xrightarrow{\text{total}} P(\Lambda)$, assigning to each transition a probability distribution over the target locations.[1] Here and in the sequel, $P(M)$ denotes the set of probability distributions over the finite set M.

- A family $g = (g_t)_{t \in \text{Trans}}$ assigning to each transition a *transition guard* enabling that transition, where the transition guard is an arithmetic predicate with free variables in R that may involve non-linear arithmetic and transcendental functions, as can be seen in Figure 2.2.

- A family $\text{asgn} = (\text{asgn}_{t,\lambda'})_{t \in \text{Trans}, \lambda' \in \Lambda}$ assigning to each transition and each target location an *assignment* which is defined by means of a predicate over variables in R and R', where $R' = \{x'_1, \ldots, x'_n\}$ denotes primed variants of the state components in R. Undecorated state components $x \in R$ refer to the state immediately before the transition, while the primed variant $x' \in R'$ refers to the state immediately thereafter such that the predicates define pre-post-relations. The assignment predicates are potentially non-linear arithmetic predicates over the continuous variables and may involve transcendental functions; Figure 2.2 provides examples of such assignment predicates.

  To maintain the desired separation between the resolution of non-determinism and random transitions, we demand that assignments are defined for each state satisfying the guard, i.e. require $g_t \Rightarrow \exists x'_1, \ldots, x'_n : \text{asgn}_{t,\lambda'}$ to be valid for each $t \in \text{Trans}$ and each $\lambda' \in \Lambda$ with $p(t)(\lambda') > 0$.

- A family $\text{ode} = (\text{ode}_\lambda)_{\lambda \in \Lambda}$ assigning to each location $\lambda \in \Lambda$ a *flow* $\text{ode}_\lambda : \mathbb{R}^n \xrightarrow{\text{total}} \mathbb{R}^n$ which describes the continuous evolution while residing in λ by means of a vector field, constraining the evolution to solutions of the ordinary differential equation $\frac{d\vec{x}}{dt} = \text{ode}_\lambda(\vec{x})$. For technical reasons induced by the constraint solving mechanisms for ODEs, we assume in the sequel that the automaton follows each individual flow for a total duration of at most a given $\Delta > 0$, thereafter being forced into a stutter step

[1] W.l.o.g., distributions range over the complete set Λ: unconnected locations and locations connected with probability 0 are indistinguishable w.r.t. probabilistic reachability.

before resuming the flow. Note that due to stuttering, this assumption does not prevent the automaton from residing in λ for an arbitrarily long duration, to the extent permitted by the invariant.

- A family $\text{inv} = (\text{inv}_\lambda)_{\lambda\in\Lambda}$ assigning to each location $\lambda \in \Lambda$ an *invariant* inv_λ which is a box in $\mathbb{R}^n$, i.e. specifies for each $x_i \in R$ an interval I_{x_i} of finite width that the continuous evolution may not leave while residing in λ.

- A family $\text{init} = (\text{init}_\lambda)_{\lambda\in\Lambda}$ of *initial state predicates*, where each init_λ is an arithmetic predicate over R which constrains the valuations of the continuous state components when control resides *initially* in the discrete location λ.[2]

Given a transition $\text{tr} \in \text{Trans}$, A has a tr-*jump* from state $(\lambda, \vec{x}) \in \Lambda \times (R \xrightarrow{\text{total}} \mathbb{R})$ to state $(\lambda', \vec{x}') \in \Lambda \times (R \xrightarrow{\text{total}} \mathbb{R})$ if $s(\text{tr}) = \lambda$ and $\vec{x} \models g_{\text{tr}}$ and if $\vec{x}, \vec{x}' \models \text{asgn}_{\text{tr},\lambda'}$ or $p(\text{tr})(\lambda') = 0$.[3] Here, $\vec{x}, \vec{x}' \models \text{asgn}_{\text{tr},\lambda'}$ denotes that $\text{asgn}_{\text{tr},\lambda'}$ is satisfied when $\vec{x}$ is substituted for the variables in R and $\vec{x}'$ is substituted for the variables in R' or alternatively . Reflecting the random event of selecting a target location entailed in the transition, the *probability of the* tr-*jump* from $(\lambda, \vec{x})$ to $(\lambda', \vec{x}')$ is $\text{pr} = p(\text{tr})(\lambda')$. In this case, we write $(\lambda, \vec{x}) \to^{\text{pr}}_{\text{tr}} (\lambda', \vec{x}')$.

A has a continuous *flow* from $(\lambda, \vec{x}) \in \Lambda \times (R \xrightarrow{\text{total}} \mathbb{R})$ to $(\lambda', \vec{x}') \in \Lambda \times (R \xrightarrow{\text{total}} \mathbb{R})$ if $\lambda = \lambda'$ and if there is a continuous evolution in location λ of duration at most Δ which leads from $\vec{x}$ to $\vec{x}'$, i.e. if there is a duration $t \in]0, \Delta]$ such that

$$\exists F : [0,t] \xrightarrow{\mathscr{C}^1} \mathbb{R}^n : \left(\begin{array}{ll} & F(0) = \vec{x} \\ \wedge & F(t) = \vec{x}' \\ \wedge & \forall \delta \in [0,t] : \frac{\mathrm{d}F}{\mathrm{d}t}(\delta) = \text{ode}_\lambda(F(\delta)) \\ \wedge & \forall \delta \in [0,t] : F(\delta) \in \text{inv}_{s(\text{tr})} \end{array} \right) ,$$

where $A \xrightarrow{\mathscr{C}^1} B$ denotes the continuously differentiable functions with domain A and image $\subseteq B$. In this case, we write $(\lambda, \vec{x}) \to_t (\lambda', \vec{x}')$.

A *run* of A is a sequence $(\lambda_0, \vec{x}_0) \to_{t_1} (\lambda_1, \vec{x}_1) \to^{\text{pr}_2}_{\text{tr}_2} \ldots (\lambda_n, \vec{x}_n)$ of flows and jumps. It need not alternate between flows and jumps, but may well chain multiple jumps or multiple flows in a row, thus supporting stuttering within flows as well as permitting multiple jumps at the same time instant.The probability $Pr(r)$ of a run r is the product of the probabilities of the jumps incorporated, i.e. for the above run it is $\prod_{i=1}^{n} \text{pr}_i$ with $\text{pr}_i = 1$ whenever the step is a continuous evolution and the probability of the corresponding jump otherwise. Given a run, we call its projection to the states visited, i.e. the sequence $\langle(\lambda_0, \vec{x}_0), (\lambda_1, \vec{x}_1), \ldots, (\lambda_n, \vec{x}_n)\rangle$ its *trace*. Notice the relationship between the measure Pr and *Prob* or $\mathscr{P}$, defined for the previous models (unlike Pr, notice that $\mathscr{P}$ did not distinguish between the probability associated to events in discrete or continuous time).

[2] A discrete location λ not to be taken initially takes the predicate $\text{init}_\lambda = \text{false}$.

[3] The alternative $p(\text{tr})(\lambda') = 0$ completes the transition system with default transitions of probability 0 in cases where no explicit transition is available, as arcs with probability 0 are generally not drawn in PHA and thus naturally associated with the unsiatisfiable transition predicate false. This completion is just a technicality which avoids many case distinctions in the subsequent development.

Note that beyond the stochastic uncertainty of the PHA dynamics, which is covered by the random events in transitions, there are two points in a step where uncertainty modelled as non-determinism enters. These are in the selection among the set of enabled transitions, which need not be a singleton, and among the continuous successor state in the assignment to the continuous variables. As discussed above, we assume that the dynamics is controlled by a *decision maker* or *policy (scheduler, adversary)* resolving the non-determinism based on (complete) observation of the current state and history. Based on the trajectory exhibited so far, such a policy can decide

1. whether a jump shall be performed, if one is enabled, or whether and for how long the current continuous evolution shall proceed,
2. in case a jump is performed, which of the enabled transitions $\mathrm{tr} \in \mathrm{Trans}$ shall be taken, and
3. depending on the probabilistically generated random event, which of the possible updates of continuous variables pertaining to this event shall be performed.

In the remainder, we shall assume that these decisions depend deterministically on the current trajectory prefix, i.e. that the permissible adversaries are Markovian deterministic policies [10], also known as step-dependent schedulers. Notice the relationship between this notion of policy and that introduced for DTSHS. Consequently, a policy consists of

1. a start state selected amongst the permissible initial states $\{(\lambda, \vec{x}) \mid \vec{x} \models \mathrm{init}_\lambda\}$ and
2. a mapping from sequences of sampling points to either a duration $t \leq \Delta$ of the next flow or a transition $\mathrm{tr} \in \mathrm{Trans}$ to be taken plus a Λ-indexed family of assignments to the continuous variables (one assignment for each outcome of the random event, thus covering the reaction of the policy to the random event).

I.e., with $S = \Lambda \times (R \xrightarrow{\text{total}} \mathbb{R})$ being the state set of the PHA, a policy is a pair of an initial state and a total mapping $\mathrm{Pol} : S^* \xrightarrow{\text{total}} \left( ]0, \delta] \cup \left(\mathrm{Trans} \times (\Lambda \xrightarrow{\text{total}} (R \xrightarrow{\text{total}} \mathbb{R})) \right) \right)$ from histories of the system[4] to actions[5] with the side condition that the moves suggested by the policy are feasible in the PHA. The latter amounts to demanding that whenever $(\lambda, \vec{x}) \in S$ is the last element of the sequence $s \in S^*$ then

$$\exists (\lambda', \vec{x}') \in S : (\lambda, \vec{x}) \rightarrow_t (\lambda', \vec{x}') \text{ if } \mathrm{Pol}(s) = t \in]0, \Delta]$$

and

$$\forall \lambda' \in \Lambda : (\lambda, \vec{x}) \rightarrow_{\mathrm{tr}}^{p(\mathrm{tr})(\lambda')} (\lambda', \mathrm{pa}(\lambda')) \text{ if } \mathrm{Pol}(s) = (\mathrm{tr}, \mathrm{pa}) \ .$$

[4] Histories take the form of finite state sequences, i.e. are elements of S^*.

[5] As said, actions take the form of either waiting, in which case the strategy yields a delay in the interval $]0, \delta]$, or a transition in Trans. In the latter case, possible non-determinism in the assignments has to be resolved individually for each possible random transition variant, which is catered for by assigning to each transition variant $\lambda \in \Lambda$ an assignment in $R \xrightarrow{\text{total}} \mathbb{R}$ of the variables in R.

When analysing PHA, we are generally interested in the probability of satisfying a specification under an optimal policy, where optimality may, depending on the application context, mean maximization or minimization of the probability of satisfaction. A typical example would be a safety property as a specification, where minimization of the satisfaction probability can be used for determining whether a given safety margin (in the sense of a requirement stating minimal requested probability of satisfaction) would be met under any policy, i.e. under all possible use cases of an open system, and where maximization can be used for answering the question of existence of a strategy delivering safety exceeding the safety margin. For step-bounded properties, automatic verification tools discharging such proof obligations symbolically by reduction to a stochastic formula over an arithmetic theory and by appropriate solvers have been developed in the AVACS project [43, 44, 45].

Extensions. As the finite probabilistic branching supported by the PHA model is quite restrictive, necessitating indirect and manually approximated modelling of many real-world phenomena, extensions incorporating continuously distributed probabilistic branching in jumps as well as stochastic differential equations in flows seem natural. The extension to continuous distributions in branching has been investigated in [42], there being addressed by automatic abstraction to finite probabilistic branching plus infinite non-deterministic branching, while the extension also involving stochastic ODEs has been investigated in [78], there being analyzed by means of dynamic logic. As all these extensions incorporate non-deterministic branching, the latter model can be considered close to resulting in a unification between PHA and GSHS.

2.5 Stochastic Max-Plus Models

In this section we discuss probabilistic classes of Max-Plus Linear (MPL) models. In the previous sections we discussed that stochastic hybrid models encompass both continuous dynamics (within a domain), and discrete events associated with discontinuous resets (in between domains). With focus on this latter class of behaviors, MPL systems are also discrete-event models, namely the state variable counts the (real) time associated to an event counter. Deterministic MPL models [8, 55] are widely used in process control and manufacturing and in traffic management and optimization. Their main feature is the ability to model event synchronization thanks to the operations (max, plus) that characterize their structure.

We are interested in working on extensions of these models to embed elements of stochasticity. In this Section we survey definitions of Stochastic MPL models in the literature, and sketch a few current results.

Max-Plus Algebra and Deterministic MPL Models. We define $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$. The max-plus addition and multiplication are denoted by $\oplus$ and $\otimes$ respectively. The semantics of those operators are:

$$x \oplus y = \max(x, y), \text{ and } x \otimes y = x + y,$$

where $x, y \in \mathbb{R}_{\max}$. For matrices, we have

$$[A \oplus B]_{i,j} = [A]_{i,j} \oplus [B]_{i,j},$$
$$[A \otimes C]_{i,j} = \bigoplus_{k=1}^{n} [A]_{i,k} \otimes [C]_{k,j},$$

where $A, B \in \mathbb{R}_{\max}^{m \times n}$ and $C \in \mathbb{R}_{\max}^{n \times p}$. Notice the analogy between $\oplus$ and $\otimes$ with addition and multiplication respectively in in standard vector and matrix algebra. We introduce e and E as the vector with every component equal to 0 and as the max-plus identity matrix, respectively.

A deterministic Max-Plus Linear system is described by the following dynamics:

$$x(k) = A(k) \otimes x(k-1) \oplus B(k) \otimes u(k), \tag{2.6}$$
$$y(k) = C(k) \otimes x(k), \tag{2.7}$$

where $k \in \mathbb{N}$ denotes an event counter, whereas $x(k)$ is an n-dimensional vector, $u(k)$ is an m-dimensional vector, $y(k)$ is an p-dimensional vector, $\{A(k)\}$ is a sequence of square matrices of size $n \times n$, $\{B(k)\}$ is a sequence of matrices of size $n \times m$, and $\{C(k)\}$ is a sequence of matrices of size $p \times n$. The elements of the three matrices are in $\mathbb{R}_{\max}$.

Classes of Stochastic MPL Models

Probabilistically Switching MPL Models. The contribution in [94] Introduces a class of fully-observable, autonomous Stochastic MPL. In other words, the underlying assumption is that $C(k) = E$, and $u(k) = 0, \forall k$. The underlying assumption for these models is that the state matrix is time-dependent.

Definition 11 (Probabilistically Switching MPL Models)**.** *Consider system* (2.6)-(2.7)*, with* $C(k) = E$ *and* $u(k) = 0, \forall k$*. The sequence* $\{A(k) : k \in \mathbb{N}\}$ *takes its values in a fixed set of Max-Plus matrices* $\mathscr{A} = \{A_1, \ldots, A_L\}$*, in which each matrix* A_i *is i.i.d. and associated to a non-zero probability, for* $1 \leq i \leq L$*, and* $\sum_{i=1}^{L} P(A(k) = A_i) = 1$*.*

Asymptotic properties of this class of models are studied exploiting their algebraic properties and ergodic theory.

Randomly Switching MPL Models. The previous class of Stochastic MPL models is a particular case of that in [93]. In this work the authors introduce the class of Randomly Switching MPL Models. These models can be of two types.

Definition 12. *[Type-1 Randomly Switching MPL Model] Consider system* (2.6)-(2.7)*, with matrices* $A(k), B(k), C(k)$ *belong to a discrete set of modes* $\ell(k)$*, where* L *denotes the (finite) cardinality of this set.*
Define $w(k) = \begin{bmatrix} \ell(k-1) & x^T(k-1) & u^T(k) & v^T(k) \end{bmatrix}^T$*, then for any* $m \in \{1, \ldots, L\}$ *there*

exist matrices $S_{i,m}$, vectors $\alpha_{i,m}$, $s_{i,m}$ and scalars $\beta_{i,m}$ such that the conditional probability P of mode $\ell(k)$ given vector $w(k)$ can be expressed as

$$P(\ell(k)|w(k)) = \alpha_{i,\ell(k)}^T w(k) + \beta_{i,\ell(k)},$$
$$\text{if } w(k) \in \Gamma_{i,\ell(k)} \text{ for } i = 1, \ldots, n_{\ell(k)},$$

where $\Gamma_{i,\ell(k)} = \{w(k) : S_{i,\ell(k)} w(k) \leq s_{i,\ell(k)}\}$, and the sets $\Gamma_{i,\ell(k)}$ are such that

$$\bigcup_{i=1}^{n_m} \Gamma_{i,m} = \mathbb{R}^{n_w} \text{ and } int(\Gamma_{i,m}) \cap int(\Gamma_{j,m}) = \emptyset \text{ for } i \neq j,$$

where $int(\cdot)$ denotes the interior of a set.

In the previous definition, the probability of switching to mode $\ell(k)$ given ($\ell(k-1)$, $x(k-1)$, $u(k)$, $v(k)$) (namely mode, state, and controls at the previous time) is denoted by $P(\ell(k)|\ell(k-1), x(k-1), u(k), v(k))$. Here $v(k)$ is an additional control vector. There is a difference between $u(k)$ and $v(k)$: the first is used to control the continuous states and the modes, whereas vector $v(k)$ is used to control exclusively the modes. For any given $\ell(k) \in \{1, \ldots, L\}$, the probability P is piecewise affine on polyhedral sets in the variables $\ell(k-1)$, $x(k-1)$, $u(k)$, $v(k)$.

The second type of Randomly Switching MPL models is introduced next.

Definition 13 (Type-2 Randomly Switching MPL Model)**.** *Consider system* (2.6)-(2.7)*, with matrices $A(k), B(k), C(k)$ belong to a discrete set of modes $\ell(k)$, where L denotes the (finite) cardinality of this set. The mode $\ell(k) = m$ if*

$$z(k) = \begin{bmatrix} \ell(k-1) & x^T(k-1) & u^T(k) & v^T(k) & d(k) \end{bmatrix}^T \in \Omega_m,$$

where $d(k) \in [0,1]$ is a uniformly distributed scalar signal, and where $\Omega_m = \bigcup_{j=1}^{n_m} \Omega_{m,j}$ in which $\Omega_{m,j}$ are closed convex polyhedra in the variable $z(k)$ (i.e. given by a finite number of linear inequalities) with non-overlapping interiors:

$$\Omega_{m,j} = \{z(k) : R_{m,j} z(k) \leq r_{m,j}\}, \text{ for } j = 1, 2, \ldots, n_m.$$

The role of $v(k)$ in the Type-2 Randomly Switching MPL Model is the same with the role of $v(k)$ in the Type-1 Randomly Switching MPL Model. The authors have shown in [93] that the two types of Randomly Switching MPL system are equivalent and therefore model the same class of probabilistic discrete event systems. Stability conditions and control synthesis by Model Predictive Control is studied by the same authors.

MPL Models with Random Matrices. In [40, 52], the authors introduce yet another class of Stochastic MPL models.

Definition 14 (MPL Models with Random Matrices)**.** *An MPL Model with Random Matrices is a system* (2.6)-(2.7)*, with matrices $A(k), B(k), C(k)$ made up of elements that are i.i.d. random variables with values in $\mathbb{R}_{max}$.*

Let us recall a few theoretical results from [40, 52]. In [52], the authors determine upper lower bounds of the Lyapunov exponent. The asymptotic growth rate of $x(k)$ is defined via the limits $\lim_{k\to\infty} \frac{x_j(k)}{k}$, for $1 \leq j \leq n$, provided the limits exist. If the above limits exist and all have the same value, then this value, usually denoted as $\lambda = \lambda(\{A(k)\})$, is called the Lyapunov exponent of the sequence $\{A(k)\}$. The authors assume that the stochastic MPL system is a stationary, ergodic sequence of integrable random variables and almost surely regular. Also, the authors propose an implementable procedure to compute upper lower bounds of the Lyapunov exponent.

With reference to Definition 11, in [94], the authors use an autonomous, fully-observable Stochastic MPL System where one of the precedence graph of the matrix is a basic sunflower graph. In a sunflower graph, every node has precisely one predecessor and the graph contains precisely one circuit. If the only circuit in a sunflower graph has length one, the graph is called a basic sunflower graph. [94] provides a characterization of Lyapunov exponent for a specific class of Stochastic MPL Systems.

Now, let us consider the contribution in [40]. Let $\mathscr{S}_{\text{mpns}}$ denote the set of max-plus-nonnegative-scaling functions, i.e., functions f of the form

$$f(z) = \max_{i=1,\ldots,m} (\tau_{i,1} z_1 + \cdots + \tau_{i,n} z_n + \xi_i),$$

with variable $z_i \in \mathbb{R}_{\max}$, constant coefficients $\tau_{i,j} \geq 0$, $\xi_i \in \mathbb{R}$ and $m \in \mathbb{N}$. In the sequel, we stress that f is a function of $z = \begin{bmatrix} z_1 & z_2 & \ldots & z_n \end{bmatrix}^T$ by writing $f \in \mathscr{S}_{\text{mpns}}(z)$. This work deals with the following models, a subclass of those in Definition 14.

Definition 15 (msns Stochastic MPL Models)**.** *Consider system* (2.6)-(2.7)*, where the entries of the system matrices belong to $\mathscr{S}_{mpns}$, i.e. $A(k) \in \mathscr{S}_{mpns}^{n\times n}(e(k))$, $B(k) \in \mathscr{S}_{mpns}^{n\times n_u}(e(k))$, $C(k) \in \mathscr{S}_{mpns}^{n_y\times n}(e(k))$, where $e(k)$ is a stochastic variable with a certain probability distribution.*

The variable $e(k)$ models the noise that leads to perturbations of the system matrices. The value of τ represents the effect of noise to system matrices. In [40], the authors determine a controller of this Stochastic MPL system by using a Model Predictive Control (MPC). The main contribution of the paper is on decreasing the computational complexity of the optimization problem by approximating the calculation of stochastic integrals.

2.6 Relations between models

Due to space limitations and possibly wide differences in semantical content, this survey may have not included all known and used models classes that present probabilistic and hybrid dynamics. The survey has instead focused on models with clear semantical relationships, and with potential to be integrated and to host the analysis, verification, and control synthesis techniques that are to be developed within the project. We now recapitulate the main classes of models introduced or referenced in this Section:

- DTMC, discussed in Section 2.1,

- PA, discussed in Section 2.1 and related to DTMC,
- MDP have been hinted at in Section 2.1 within PA, and discussed in Section 2.2 within DTSHS,
- CTMC, discussed in Section 2.1,
- IMC, discussed in Section 2.1 in relationship to CTMC,
- DTSHS, discussed in Section 2.2,
- GSHS, discussed in Section 2.3,
- PDMP have been briefly discussed in 2.3 with regards to GSHS and in Section 2.4 in relationship to PHA,
- SDP have been mentioned in Section 2.3 with regards to GSHS,
- SDE have been discussed in Section 2.3 in relationship to GSHS,
- PHA, discussed in Section 2.4,
- ODE have been mentioned in Section 2.4 in relationship to PHA, and in Section 2.3 in relationship to PDMP, and
- SMPL (and the following different formulations: PSMPL, RSMPL, MPLRM, and msnsSMPL), discussed in Section 2.5.

We now elaborate the discussion on the relationship between the introduced models. A few relations, particularly when dealing with simpler models, appear obvious from the model definitions, whereas others need to be detailed.

With regards to the first case, let us qualitatively denote with $\subset$ a containment relationship with regards to model semantics – in other words, this means that one model can be obtained as a particular instance of the more general one. It is then possible to observe that

- DTMC $\subset$ PA,
- MDP $\subset$ PA,
- PA $\subset$ DTSHS,
- CTMC $\subset$ IMC and PDMP,
- PDMP $\subset$ GSHS,
- SDP $\subset$ GSHS,
- SDE $\subset$ SDP, and
- ODE $\subset$ PHA and PDMP.

Next, we discuss less obvious relations and explicitly develop connections between a few models.

PHA and GSHS. In comparison to GSHS (Sect. 2.3), PHA of Section 2.4 add a second form of choice dynamics, namely non-deterministic choice in both the selection among competing transitions and the computation of the continuous successor state. They are, however, more confined in their continuous dynamics: PHA do neither provide the transition rates, nor stochastic resets, nor the SDE present in GSHS.

Since the finite probabilistic branching supported by the PHA model is quite restrictive, necessitating indirect and manually approximated modelling of many real-world phenomena, extensions incorporating continuously distributed probabilistic branching in jumps as well as stochastic differential equations in flows have been suggested. The extension to continuous distributions in branching has been investigated in [42], while the extension also involving stochastic ODE has been investigated in [78].

From DTSHS to DTMC. We introduce an approximation scheme to abstract a concrete DTSHS model into a DTMC, which is based on a discretization procedure discussed in [2]. The procedure focuses on the autonomous case ($\mathscr{U} = \emptyset$).

We perform the discretization over a compact set $A \in \mathscr{B}(\mathscr{S})$, given by $A = \cup_{q\in\mathscr{Q}}\{q\} \times A_q$ with $A_q \in \mathscr{B}(\mathbb{R}^{n(q)})$. (As a special case, the technique can be used when $A = \mathscr{S}$, provided the state space is bounded.) The size of the continuous state space within A is measured by $\lambda := \max_{q\in\mathscr{Q}} \mathscr{L}(A_q)$, where $\mathscr{L}(A_q) < \infty$ denotes the finite Lebesgue measure of the set $A_q \subset \mathbb{R}^{n(q)}$. Assume for simplicity that $A_q \neq \emptyset$ for all $q \in \mathscr{Q}$. Since A is compact, we can introduce a finite partition of each compact set $A_q \subset \mathbb{R}^{n(q)}$, $q \in \mathscr{Q}$, by taking $A_q = \cup_{i=1}^{m_q} A_{q,i}$, where $A_{q,i}$, $i = 1,\ldots,m_q$, are pairwise disjoint Borel sets $A_{q,i} \in \mathscr{B}(\mathbb{R}^{n(q)})$, with $A_{q,i} \cap A_{q,j} = \emptyset$, $\forall i \neq j$. Denote with $\delta_{q,i}$ the diameter of the set $A_{q,i}$, that is $\delta_{q,i} = \sup\{\|x - x'\| : x, x' \in A_{q,i}\}$, and define the *grid size parameter* by $\delta := \max_{i=1,\ldots,m_q, q\in\mathscr{Q}} \delta_{q,i}$.

The collection of sets $\mathscr{G} := \{G_{q,i} := \{q\} \times A_{q,i},\ i = 1,\ldots,m_q,\ q \in \mathscr{Q}\}$ represents a partition of the safe set A. For each element $G_{q,i}$ of the partition, we select a representative point $(q, v_{q,i}) \in G_{q,i}$. The set $A_\delta := \{(q, v_{q,i}), i = 1,\ldots,m_q,\ q \in \mathscr{Q}\}$ is the discretized version of the safe set A. We denote with $\xi : A \to A_\delta$ the map that associates to $s \in G_{q,i} \subset A$ the corresponding discrete state $(q, v_{q,i}) \in A_\delta$, and with $\Xi : A_\delta \to \mathscr{G}$ the set-valued map that associates to $(q, v_{q,i}) \in A_\delta$ the set $G_{q,i}$ to which $(q, v_{q,i})$ belongs.

We next introduce the state space $\mathscr{Z}_\delta$ and the transition probability function $T_\delta : \mathscr{Z}_\delta \times \mathscr{Z}_\delta \to [0,1]$ of a stochastic automaton that approximates the original DTSHS for the purpose of probabilistic invariance computation. The state space of the stochastic automaton is defined as $\mathscr{Z}_\delta := A_\delta \cup \{\phi\}$, where ϕ is a discrete state representing the set of all states in the hybrid state space $\mathscr{S}$ that are outside the safe set A. Notice that the compactness assumption on A ensures that the set $\mathscr{Z}_\delta$ is finite.

The transition probability function $T_\delta : \mathscr{Z}_\delta \times \mathscr{Z}_\delta \to [0,1]$ is defined as follows:

$$T_\delta(z'|z) = \begin{cases} T_s(\Xi(z')|z), & \text{if } z' \in A_\delta \text{ and } z \in A_\delta \\ 1 - \sum_{\bar{z}\in A_\delta} T_s(\Xi(\bar{z})|z), & \text{if } z' = \phi \text{ and } z \in A_\delta \\ 1, & \text{if } z' = z = \phi \\ 0, & \text{if } z' \in A_\delta \text{ and } z = \phi, \end{cases} \tag{2.8}$$

and satisfies $\sum_{z' \in \mathscr{Z}_\delta} T_\delta(z'|z) = 1$, for all $z \in \mathscr{Z}_\delta$. Note that ϕ is an absorbing state and the probability that the stochastic automaton evolves from a safe state $z \in A_\delta$ to a safe state $z' \in A_\delta$ is defined as the probability that the original DTSHS will enter the safe set $\Xi(z') \subset A$ in one time step starting from z.

The execution during the time horizon $[0, N]$ of the stochastic finite automaton associated with the initial condition $z_0 \in \mathscr{Z}_\delta$ is a Markov chain $\{\mathbf{z}(k),\ k \in [0, N]\}$ defined on the probability space $(\mathscr{Z}_\delta^{N+1}, \sigma(\mathscr{Z}_\delta^{N+1}), Prob_{\delta,z_0})$, where $\sigma(\mathscr{Z}_\delta^{N+1})$ is the σ-algebra associated to $\mathscr{Z}_\delta^{N+1}$, and the probability measure $Prob_{\delta,z_0}$ is uniquely defined by the initial condition z_0 and the transition probability function T_δ.

In [2], the discretization is employed as a method for approximate model checking of stochastic hybrid systems with provable approximation guarantees. Focusing on the probabilistic invariance problem for discrete time stochastic hybrid systems, the model is first approximated by a finite state Markov chain. The approximating chain is then model checked for probabilistic invariance. Under certain regularity conditions on the transition and reset kernels governing the dynamics of the stochastic hybrid system, the invariance probability computed using the approximating Markov chain is shown to converge to the invariance probability of the original stochastic hybrid system, as the grid used in the approximation gets finer. A bound on the convergence rate is also provided.

From GSHS to CTMC-like models. We sketch an approximation scheme from [4], aimed at transforming a GSHS into a SHS without forced transitions due to spatial guards. If the resets are disregarded, this latter model can be regarded as a SDP [47], that is as a generalization of CTMC with continuous spatial components. The forced switching mechanisms are replaced by spontaneous transitions with state-dependent transition intensities (jump rates). The resulting switching diffusion process with random hybrid jumps is shown to converge in distribution to the original stochastic hybrid system execution.

Consider the GSHS system $\mathscr{S}_g$ in Definition 8. The guard set of $\mathscr{S}_g$ within mode $q \in \mathscr{Q}$ is made up of $\gamma_{qq'} \subset \mathbb{R}^{n(q)}$, $q' \in \mathscr{Q}$, $q' \neq q$. Assume that each set $\gamma_{qq'}$ can be expressed as a zero sub-level set of a continuous function $h_{qq'} : \mathbb{R}^{n(q)} \to \mathbb{R}$:

$$\gamma_{qq'} = \{x \in \mathbb{R}^{n(q)} : h_{qq'}(x) \leq 0\}.$$

Pick a small enough $\delta > 0$, and by the continuity of $h_{qq'}$, introduce the sets

$$\gamma_{qq'}^{-\delta} = \{x \in \mathbb{R}^{n(q)} : h_{qq'}(x) \leq -\delta\} \subseteq \gamma_{qq'} \subseteq \gamma_{qq'}^{\delta} = \{x \in \mathbb{R}^{n(q)} : h_{qq'}(x) \leq \delta\}.$$

For any $q \in \mathscr{Q}$, define the set of functions $\lambda_{qq'}^{\delta} : \mathbb{R}^{n(q)} \to \mathbb{R}^+$, $q' \in \mathscr{Q}$, $q' \neq q$,

$$\lambda_{qq'}^{\delta}(x) = \begin{cases} \left(\dfrac{1}{d(x, \gamma_{qq'}^{-\delta})} - \dfrac{1}{\sup\limits_{y : h_{qq'}(y) = \delta} d(y, \gamma_{qq'}^{-\delta})} \right) \wedge \left(\dfrac{1}{\sup\limits_{y : h_{qq'}(y) = 0} d(y, \gamma_{qq'}^{-\delta})} \right), & x \in \gamma_{qq'}^{\delta} \\ 0, & x \in \mathbb{R}^{n(q)} \setminus \gamma_{qq'}^{\delta} \end{cases}$$

where $a \wedge b = \min\{a, b\}$, whereas $d(z, A) = \inf_{y \in A} ||z - y||, z \in \mathbb{R}^{n(q)}, A \subset \mathbb{R}^{n(q)}$.

We associate to $\mathscr{S}_g$ a new stochastic hybrid system $\mathscr{S}_\delta$, which is made up of the elements of $\mathscr{S}_g$, except for the following:

- the spatial guards set is empty, $\Gamma = \emptyset$,
- the transition intensity function Λ, whose domain of definition is $\mathscr{S} \setminus \Gamma \times \mathscr{Q}$, is replaced by $\Lambda^\delta : \mathscr{S} \times \mathscr{Q} \to \mathbb{R}^+$ given by $\Lambda^\delta((q, x), q') := \lambda^\delta_{qq'}(x) + \lambda_{qq'}(x)$ where for any $q' \neq q \in \mathscr{Q}$, the original jump rate $\lambda_{qq'}(\cdot)$ is extended to $\mathbb{R}^{n(q)}$ by setting it to zero over Γ_q, and
- the stochastic reset kernel $R^{\Lambda^\delta} : \mathscr{B}(\mathbb{R}^{n(\cdot)}) \times \mathscr{Q} \times \mathscr{S} \to [0, 1]$ associated with Λ^δ is given by $R^{\Lambda^\delta}(C_{q'}|q', (q, x)) = R^\Lambda(C_{q'}|q', (q, x)) + R^\Gamma(C_{q'}|q', (q, x))$, for any Borel set $C_{q'}$ of $\mathbb{R}^{n(q')}$, where the original stochastic reset kernels $R^\Lambda(C_{q'}|q', \cdot)$ and $R^\Lambda(C_{q'}|q', \cdot)$ are extended to $\mathscr{S}$ by setting them to zero outside their original domain of definition.

Notice that the conclusions in Proposition 1 hold true also for the SHS $\mathscr{S}_\delta$. The extended generator $\mathscr{L}_\delta$ of $\mathscr{S}_\delta$ can be derived in a similar way as for that for $\mathscr{S}_g$, but has no condition on the points of the guard set. Its domain $\mathscr{D}(\mathscr{L}_\delta)$ is the set of functions $f \in C^2_b(\mathscr{S})$. This implies that $\mathscr{D}(\mathscr{L}_g) \subseteq \mathscr{D}(\mathscr{L}_\delta)$, for $\delta > 0$.

Let us formally show that, as $\delta \to 0$, the sequence of stochastic processes $\{\mathbf{s}_\delta(t)\}_{\delta > 0}$ converges, in some sense, to $\mathbf{s}(t)$, for any $t \geq 0$. The concept of extended generator can be useful in showing that a sequence of Markov processes converges to a given Markov process. Qualitatively, given a sequence of $\mathscr{S}$-valued processes $\{\mathbf{X}_n\}_{n \geq 1}$ and a process $\mathbf{X}$, with extended generators $(A_n, \mathscr{D}(A_n))$ and $(A, \mathscr{D}(A))$ respectively, to prove that $\mathbf{X}_n \Rightarrow \mathbf{X}$ (convergence in the weak sense), it is sufficient to show that for all functions $f \in \mathscr{D}(\mathscr{A})$, there exist $f_n \in \mathscr{D}(\mathscr{A}_n)$, such that $f_n \to f$ and $A_n f_n \to Af$. The following fact, needed in Theorem 2, is verified:

Theorem 1 (Compact Containment Condition)**.** *Consider the GSHS $\mathscr{S}_g$, the SHS $\mathscr{S}_\delta$, and their corresponding unique global solutions $\mathbf{s}(t)$ and $\mathbf{s}_\delta(t), t \geq 0$. The stochastic processes $\mathbf{s}_\delta(t)$ are such that, for any $\epsilon > 0, N > 0$, there exists a compact set $K_{\epsilon,N} \subset \mathscr{S}$ such that*

$$\liminf_{\delta \downarrow 0} \mathscr{P}\left[\mathbf{s}_\delta(t) \in K_{\epsilon,N}, \forall\, 0 \leq t \leq N\right] \geq 1 - \epsilon.$$

Similarly for the stochastic process $\mathbf{s}(t)$. □

Given a sequence of entities $\{c_n\}_{n \geq 1}$ and a scalar c, let us denote as $\lim^\star_n c_n = c$ the conditions $\lim_{n \to \infty} c_n = c$ and $\left(\vee_n \|c_n\|\right) \vee \|c\| < \infty$, where $\|\cdot\|$ is the sup norm. Similarly if the indexing parameter tends to zero ($\delta = 1/n$). A process $\mathbf{X}$ is said to be a solution of the local martingale problem for a linear operator (A, π) if $\mathscr{P} \circ \mathbf{X}(0)^{-1} = \pi$, and for each $f \in \mathscr{D}(A)$, $f(\mathbf{X}(t)) - f(\mathbf{X}(0)) - \int_0^t Af(\mathbf{X}(s))ds$ is a local martingale, $\forall t \geq 0$. In order to complete the proof of the following Theorem 2, it is necessary to raise the following

Assumption 7. *We assume the following:*

1. *given a GSHS, as in Definition 8, assume that the probabilistic reset kernels* $R^{\Gamma}(\cdot|q',(q,x))$ *are continuous in* x*, for any* $q' \neq q \in \mathcal{Q}$*, and*

2. *the local martingale problem for* $(\mathscr{L}_g, \mathscr{D}(\mathscr{L}_g))$ *is well posed, that is, it admits a* unique *solution.* □

The following theorem is based on results from [96, Theorem 4.4].

Theorem 2 (Weak Convergence of $\mathscr{S}_\delta$ to $\mathscr{S}_g$)**.** *Consider the SHS model* $\mathscr{S}_\delta$*, the GSHS model* $\mathscr{S}_g$ *under Assumption 7.1, and their associated* $\mathscr{S}$*-valued unique solution processes* $\mathbf{s}_\delta(t)$ *and* $\mathbf{s}(t), t \geq 0$*, where* $\mathbf{s}_\delta(0) = \mathbf{s}(0) = (q_0, x_0) \in \mathscr{S}$*. Consider further their extended generators* $(\mathscr{L}_\delta, \mathscr{D}(\mathscr{L}_\delta))$ *and* $(\mathscr{L}_g, \mathscr{D}(\mathscr{L}_g))$*, and conjecture that Assumption 7.2 is valid. It holds that*

- $\mathscr{L}_g \subset C_b^0(\mathscr{S}) \times C^0(\mathscr{S})$*;*
- *For all* $f \in \mathscr{D}(\mathscr{L}_g), \exists f_\delta \in \mathscr{D}(\mathscr{L}_\delta)$*, such that* $\lim_\delta^\star f_\delta = f, \lim_\delta \mathscr{L}_\delta f_\delta = \mathscr{L} f$*;*
- $\mathscr{D}(\mathscr{L})$ *is dense in* $C_b^0(\mathscr{S})$ *with respect to* $\lim^\star$.

By Theorem 1, as the approximation step $\delta \downarrow 0$*, the solution of the SHS* $\mathscr{S}_\delta$ *weakly converges to that of the GSHS* $\mathscr{S}_g$*:* $\mathbf{s}_\delta(t) \Rightarrow \mathbf{s}(t), \forall t \geq 0$. □

The obtained approximation can be useful for various purposes such as, on the computational side, simulation and reachability analysis, as well as for the theoretical investigation of the model. More generally, it is suggested that SHS which are endowed exclusively with random jumping events are *simpler* than those that present spatial forcing transitions.

From PHA to PA. An abstraction embedding PHA into PA while safely overapproximating reach probabilities has been suggested in [97, 98]. The crucial observation underlying that abstraction is that classical region-based (i.e., homomorphic existential) abstractions of hybrid automata, be it those obtained by "gridding" the continuous state space or those originating from counter-example guided abstraction refinement, leave the branching structure of the hybrid automaton intact. As they copy each concrete transition to all abstract states where the guard of the corresponding transition may be enabled, there is a one-to-many relation between concrete and abstract transitions, facilitating backpatching of probabilities in abstractions obtained from non-deterministic relaxations of the PHA. Akin to the procedure developed by Sproston et al. for timed automata [67, 86, 87], the procedure from [97, 98] first generates a non-deterministic counterpart of the PHA under investigation by stripping of all probabilities, thereby converting probabilistic into non-deterministic branching. The resulting hybrid automaton can be abstracted by standard methods, yielding a finite automaton overapproximating reachability in the hybrid automaton. Identifying the abstract transitions corresponding to orginally probabilistic branching and backpatching the according probabilities, thereby converting some of the non-deterministic branching back into probabilistic branching, this finite automaton finally is converted into a PA. This provides a PA that overapproximates reach probabilities in the original PHA. The resultant PA finally can be analyzed by standard analysis tools for PA, like the model checker PRISM [59].

From SMPL to DTSHS (work in progress). The formal connection between SMPL models (in their MPLRM definition that depends on random matrices) and DTSHS is a novel goal that is currently under investigation within MoVeS. The plan of work is the following. It is known that, given a deterministic MPL model, there exists a piece-wise affine (PWA) model with equivalent dynamics [54]. This connection has been recently exploited to compute finite (bi)simulations of MPL models in [6] for their analysis and verification. The goal is to extend this result to stochastic MPL models, where the state matrices are characterized by random variables. The DTSHS framework will accommodate for this.

From continuous-time to discrete-time models. It is of interest to briefly comment on the relationship between the continuous- and the discrete-time models presented in this Section. As argued in Section 2.1, a classical result for Markov processes allows approximating the probabilistic behavior of CTMC by its *embedded DTMC*. Such a time-discretization procedure becomes more involved when approximating a space-dependent process (such as an SDE). Time discretization procedures for certain classes of stochastic hybrid models have been discussed in [46, 64].

More elaborated techniques involve both time discretization and state-space gridding, and result in a DTMC that is easier to analyze. Weak-approximations techniques have been developed for this purpose in [66], and applied to classes of stochastic hybrid models in [80].

The role of Inputs in the presented models.

In this concluding paragraph, we discuss the role of controls. The presence of a control structure has been embedded in the following models:

1. PA, where a scheduler or an adversary has been considered as the entity resolving the non-determinism of the model,
2. MDP, where controls are studied as policies over a time horizon,
3. DTSHS, where policies over a finite time horizon have been introduced as strings of non-randomized, Markovian, state-feedback controls. However policies can in principle also act adversarially,
4. PHA, where as in PA policies are intended to resolve non-determinism and are thus quite general: policies can be unbiased (due to the environment, scheduler), adversarial (worst-case, demonic), or controlled (selected by a decision maker, angelic).
5. the framework of GSHS can also accommodate the presence of a control structure, as detailed in [19]. This extension is based on related work on the following models:
6. PDMP [25], on SDP [47, 48], and of course on SDE [76]. And finally for
7. SMPL models, which are often non-autonomous.

In the literature, the semantics of the control structure in the models has lead to a categorization of the models within MDP, DTSHS, GSHS on the one side, versus PA, PHA on the other. This difference reflects on the synthesis techniques typically developed to accommodate for the selection of the available controllers. Notice that this difference in the semantics of a control structure (or of non-determinism) reflects in later parts of the report. For instance, in Section 4 we give definition of bisimulations for controlled models, whereas in the formal verification literature these definitions have been first introduced for non-deterministic models such as Labeled Markov Processes, where non-determinism is intended as an environmental effect.

3 Model Composition

Compositional modeling techniques aim at obtaining computationally scalable analysis and control procedures, and enable a structured modeling of the system whereby in principle the global properties can be analyzed through the independent study of system components. This Section reviews compositional techniques for Markovian models such as DTMC (cfr. Section 2.1) via Concurrent MC. Furthermore, it discusses this notion for hybrid models such as PHA (cfr. Section 2.4).

3.1 Concurrent Markov Chains

This section presents some approaches to compose Markov chains in parallel. We introduce asynchronous parallel composition on probabilistic automata [85], as introduced in Section 2.1. Moreover, we introduce interactive Markov chains [57] (an extension of CTMC), and show how they can be defined in parallel. Both parallel compositions in the discrete- and continuous-time setting are inspired by the framework of communicating sequential processes (CSP) [17].

3.1.1 Composing discrete-time Markov chains

We start introducing the notion of parallel composition of Probabilistic Automata (PA). PA are formally introduced among the Markovian formalisms in Section 2.1. For countable set S, let $Dist(S)$ denote the set of distributions on S.

Definition 16 (Parallel composition of PA)**.** *For PA* $\mathscr{P}_1 = (S_1, A_1, \rightarrow_1, s_{0,1})$ *and* $\mathscr{P}_2 = (S_2, A_2, \rightarrow_2, s_{0,2})$ *and set of actions* $A \subseteq (A_1 \cap A_2) \setminus \{\tau\}$*, the* parallel composition $\mathscr{P}_1 ||_A \mathscr{P}_2 = (S, A_1 \cup A_2, \rightarrow, s_0)$ *where:* $S = S_1 \times S_2$ $s_0 = (s_{0,1}, s_{0,2})$*, and* $(s_1, s_2) \xrightarrow{\alpha} \mu$ *with* $\mu \in Dist(S)$ *whenever:*

- *if* $\alpha \in A$*, then* $\mu(t_1, t_2) = \mu_1(t_1) \cdot \mu_2(t_2)$*,*
- *if* $\alpha \in A_1 \setminus A$*, then* $\mu(t_1, t_2) = \mu_1(t_1)$ *and* $t_2 = s_2$*, and*
- *if* $\alpha \in A_2 \setminus A$*, then* $\mu(t_1, t_2) = \mu_2(t_2)$ *and* $t_1 = s_1$*.*

The symbol τ, to be explained shortly, denotes an internal action. The parallel composition of $\mathscr{P}_1$ and $\mathscr{P}_2$ is thus the product of these PA where the probability to simultaneously perform an action α in the synchronisation set A is the product of the individual probabilities in the components $\mathscr{P}_1$ and $\mathscr{P}_2$. For autonomously performed actions, this is similar except that the other component idles with unit probability. This parallel composition is symmetric. In the next Section 4 we will see that bisimulation is a congruence with respect to parallel composition, allowing a component-based comparison of complex system models.

As in principle any action of a PA may be subject to interaction with the environment, it is often useful to facilitate the possibility to make certain actions internal, i.e., no longer subject to synchronisation. In the sequel, we adopt the usual convention that τ denotes an internal action. This means that for any synchronisation set A we assume that $\tau \notin A$ (even if $\tau \in A_1$ and $\tau \in A_2$); i.e., $A \subseteq (A_1 \cap A_2) \setminus \{\tau\}$. The operation to declare certain actions internal is known as hiding.

Definition 17 (Hiding of PA). *Let PA $\mathscr{P} = (S, A, \rightarrow, s_0)$ and H a set of actions with $\tau \notin H$. Then $\mathscr{P} \setminus H$ is the PA $(S, A \setminus H, \rightarrow', s_0)$ where $\rightarrow'$ is defined by:*

- *if $s \xrightarrow{\alpha} \mu$ and $\alpha \in H$ then $s \xrightarrow{\tau}' \mu$, and*
- *if $s \xrightarrow{\alpha} \mu$ and $\alpha \notin H$ then $s \xrightarrow{\alpha}' \mu$.*

Thus any action in the set H is turned into an internal action, whereas transitions labeled with $\alpha \notin H$ remain unaffected. The target distributions of transitions are retained as is.

3.1.2 Composing continuous-time Markov chains

This section considers the parallel composition and hiding of classes of CTMC. The composition of CTMC has been an intensive topic of research in the context of process calculi; for a survey consult [58]. The main approach has been to use labelled transition systems, where transitions are labelled with pairs of actions and rates: this is an action-based extension of CTMC. The main drawback of this approach, however, is the technical complication arising from the treatment of time consumption in case of synchronisation. Technically, this amounts to the computation of the resulting rate in case two transitions labelled like (a, λ) and (a, μ) synchronise. The most natural interpretation is to require both delays to have expired before the interaction (on a) can take place. The resulting probability distribution, which is the product of two exponential distributions of rate λ and μ respectively, is however not an exponential distribution. (The product of two distribution functions amounts to the maximum of their corresponding random variables.) To overcome this problem, several solutions have been suggested that, however, either lack a clear stochastic interpretation or have a restricted applicability. In this section, we maintain the most natural stochastic interpretation (namely, the maximum of random variables) by explicitly *separating* between the advance of time and the occurrence of actions. This distinction leads to a behaviour where two distinct phases are mixed. Phases during which one or more actions occur (together with their corresponding state changes), but where no time elapses, alternate with phases where time passes, but during which no actions happen. This separation of discrete and continuous phases is similar to that in timed automata [7]. This yields a mixture of labelled transition systems and CTMC, known as *interactive Markov chains* [57], as introduced in Section 2.1, in which action-labelled and rate-labelled transitions co-exist. As for PA, action τ models internal activity, whereas all other actions represent external activities.

The main strength of IMCs is that they are compositional, as the following result elaborates.

Definition 18 (Parallel composition of IMCs)**.** *Let* $\mathscr{I}_1 = \left(S_1, A_1, \rightarrow_1, \Longrightarrow_1, s_{0,1}\right)$ *and* $\mathscr{I}_2 = \left(S_2, A_2, \rightarrow_2, \Longrightarrow_2, s_{0,2}\right)$ *be IMC. The* parallel composition *of* $\mathscr{I}_1$ *and* $\mathscr{I}_2$ *wrt. set* $A \subseteq (A_1 \cap A_2) \setminus \{\tau\}$ *of actions is defined by:*

$$\mathscr{I}_1 \,||_A\, \mathscr{I}_2 = \left(S_1 \times S_2, A_1 \cup A_2, \rightarrow, \Longrightarrow, (s_{0,1}, s_{0,2})\right)$$

where $\rightarrow$ *and* $\Longrightarrow$ *are defined as the smallest relations satisfying all of the following constraints:*

1. $s_1 \xrightarrow{\alpha}_1 s_1'$ *and* $s_2 \xrightarrow{\alpha}_2 s_2'$ *and* $\alpha \in A$, $\alpha \neq \tau$ *implies* $(s_1, s_2) \xrightarrow{\alpha} (s_1', s_2')$,
2. $s_1 \xrightarrow{\alpha}_1 s_1'$ *and* $\alpha \notin A$ *implies* $(s_1, s_2) \xrightarrow{\alpha} (s_1', s_2)$ *for any* $s_2 \in S_2$,
3. $s_2 \xrightarrow{\alpha}_2 s_2'$ *and* $\alpha \notin A$ *implies* $(s_1, s_2) \xrightarrow{\alpha} (s_1, s_2')$ *for any* $s_1 \in S_1$,
4. $s_1 \overset{\lambda}{\Longrightarrow}_1 s_1'$ *implies* $(s_1, s_2) \overset{\lambda}{\Longrightarrow} (s_1', s_2)$ *for any* $s_2 \in S_2$, *and*
5. $s_2 \overset{\lambda}{\Longrightarrow}_2 s_2'$ *implies* $(s_1, s_2) \overset{\lambda}{\Longrightarrow} (s_1, s_2')$ *for any* $s_1 \in S_1$.

The first three constraints define a form of parallel composition [17]: actions in A need to be performed by both IMC simultaneously, except for internal actions (first constraint), whereas actions not in A are performed autonomously (second and third constraint). According to the last two constraints, IMC can delay independently. This differs from timed models such as timed automata, in which individual processes typically need to synchronise on the advance of time. The memoryless property of exponential distributions justifies independent delaying: if two Markovian transitions with rates λ and μ, say, are competing to be executed, then the remaining delay of the μ-transition after the λ-transition has been taken is exponentially distributed with rate μ.

Definition 19 (Hiding of IMC)**.** *The* hiding *of IMC* $\mathscr{I} = \left(S, A, \rightarrow, \Longrightarrow, s_0\right)$ *wrt. the set* H *of actions is the IMC* $\mathscr{I} \setminus H = \left(S, A \setminus H, \rightarrow', \Longrightarrow, s_0\right)$ *where* $\rightarrow'$ *is the smallest relation defined by:*

1. $s \xrightarrow{\alpha} s'$ *and* $\alpha \notin H$ *implies* $s \xrightarrow{\alpha}' s'$, *and*
2. $s \xrightarrow{\alpha} s'$ *and* $\alpha \in H$ *implies* $s \xrightarrow{\tau}' s'$.

Hiding thus transforms α-transitions, with $\alpha \in H$, into interactive τ-transitions. All other transition labels remain unaffected. This operation is of importance for the maximal progress assumption of IMC. Turning an α-transition emanating from state s, say, into a τ-transition may change the semantics of the IMC at hand, as after hiding no Markovian transition will be ever taken in s.

A generalization of IMC where rates are parameterized functions is provided in [53], which includes parallel composition and hiding. Related work for PDMP-like processes has been reported in [88].

3.2 Concurrent Probabilistic Hybrid Automata

For probabilistic hybrid automata (PHA), it is quite straightforward to define shared-state concurrency if the components to be connected obey appropriate constraints on their signature, namely that each variable in the globally visible, shared state space is controlled by at most one component (but may be read by arbitrarily many). This condition may seem restrictive, but is quite common in hybrid modeling, where de-facto industry standards like Simulink enforce the very same constraint, and where the presence of analog switches makes simulating more liberal access policies rather straightforward.

If an omniscient policy is admitted, i.e. if the policy has introspection into and can base its strategy on the complete state space of all parallel components, concurrent execution of PHA obeying the aforementioned signature constraint can be explained by a straightforward product construction. In [44, 89], it has been argued that such shared-state concurrency corresponds to conjunction of the stochastic satisfiability modulo theory (SSMT) encodings of the individual components[1] part, such that SSMT provides a compositional representation of shared-state concurrency in PHA.

To explain this compositional encoding in more detail, we subsequently introduce the definition of a shared-state concurrency model for discrete-time probabilistic hybrid automata (see section 2.4 for a formal definition). The core idea for the composition is to put all concurrent automata into a common global state-space shared by all component automata, in which they agree on a global transition composed of individual local transitions and perform it synchronously. Communication is done by direct access to signature variables of the PHA. In order to avoid ill-formed behaviour, writing to such a variable has to be restricted. Every variable from the signature of each PHA can be read by an arbitrary number of components, but is controlled only by at most one.

Qualitatively, the concurrent PHA performs the following procedure:

1. **Selection of local transitions:** Based on the current global state, every individual automata selects an individual local transition among its enabled transitions and synchronously suggests it to the environment.

2. **Consensus on a global transition:** A consensus on a single global transition is established by combining the local transitions from each concurrent component.

3. **Selection of probabilistic transition variants:** After committing to a global transition, a probabilistic variant of every corresponding local transition is selected.

4. **Consensus on transition side effects:** A global consensus on the execution of the locally selected probabilistic variants is established by checking mutual consistency of all transition side effects. This permits multiple PHA to write the same variables, if and only if they propose the same value or —in case of non-deterministic assignments— overlapping sets of possible assignments.

[1] More precisely, concurrent composition corresponds semantically to conjunction of the quantifier-free formula bodies and to step-consistent interleaving of the quantifier prefixes encoding the component dynamics.

5. **Execution of global transition:** In case of a consensus, the transition is executed concurrently by applying the associated effects on the global state. If no consensus can be found then the system deadlocks due to inconsistent assignments in the committed transitions.

In the following, we define a formal model for this procedure and show how it can be used for probabilistic bounded reachability analysis. These sections is extracted from [89], where a more comprehensive survey of this topic can be found.

Syntax and semantics of concurrent PHA

A *system of concurrent discrete-time probabilistic hybrid automata* $\mathscr{S} = \{\mathscr{A}_1, \ldots, \mathscr{A}_n\}$ is given by a set of *discrete-time probabilistic hybrid automata*, where each probabilistic hybrid automaton $\mathscr{A}_i$ consists of the following:

- A finite set $D_i = \{d_1^i, \ldots, d_{k_i}^i\}$ of *discrete variables* spanning the discrete state space (sometimes called the locations or the modes) of the hybrid automaton by means of the Cartesian product $\prod_{j=1}^{k_i} \text{range}(d_j^i)$ of each their *finite* ranges, $\text{range}(d_j^i)$. In order to permit non-local referencing of the state variables, we require $D_i \cap D_j = \emptyset$ if $i \neq j$, i.e. that the variable names used in different concurrent automata are disjoint.

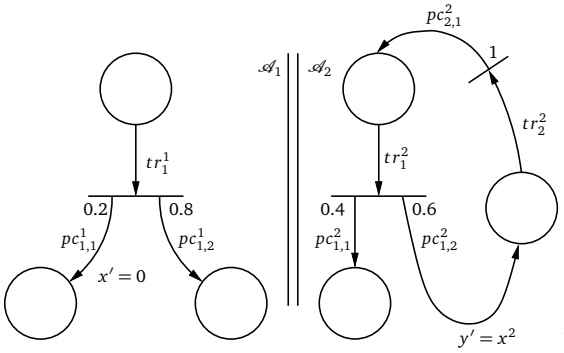
- A finite dimensional vector $R_i = \{x_1^i, \ldots, x_{m_i}^i\}$ of *continuous state components* controlled by automaton $\mathscr{A}_i$, yet visible to all others. Each continuous component x_j^i ranges over an interval $\text{range}(x_j^i) = [l_{x_j^i}, u_{x_j^i}]$ within the reals $\mathbb{R}$. Again, we demand that $R_i \cap R_j = \emptyset$ if $i \neq j$. Additionally, we require discrete variable names and continuous variable names to be disjoint, i.e. $D_i \cap R_j = \emptyset$ for all i and j.

- A predicate init_i in an arithmetic theory $\mathscr{T}$ with free variables in D_i and R_i describing the *initial state* of the automaton. For technical reasons and without loss of generality, we require that there is exactly one valuation in the set of states $\text{States}_i = \prod_{j=1}^{k_i} \text{range}(d_j^i) \times \prod_{j=1}^{m_i} \text{range}(x_j^i)$ of the automaton which satisfies init_i. Note that due to the disjointness of the local variable name spaces, this implies the existence of exactly one global initial state $s \in \prod_{i=1}^{n} \text{States}_i$ satisfying $\bigwedge_{i=1}^{n} \text{init}_i$.

- A finite family $Tr_i = \{tr_1^i, \ldots, tr_{\ell_i}^i\}$ of *symbolic transitions*.

Each symbolic transition tr_j^i comprises the following:

- A *generalized transition guard* $g(tr_j^i)$ expressing the conditions on variables required for establishing consensus on that transition. $g(tr_j^i)$ is an arithmetic predicate in the arithmetic theory $\mathscr{T}$ over variables in $D_1, \ldots, D_n$ and $R_1, \ldots, R_n$ as well as primed variants thereof, the latter representing the post-states. A transition guard states the conditions on the discrete as well as the continuous state under which the transition may be taken. Note that the guard predicate can refer to the current states and post-states of all concurrent automata in $\mathscr{S}$. It thus provides an expressive formalism supporting synchronization through global consensus.



$$\text{NChoice} = \{(tr^1_1, tr^2_1), (tr^1_1, tr^2_2)\}$$
$$\text{PChoice}((tr^1_1, tr^2_1)) = \{(pc^1_{1,1}, pc^2_{1,1}), (pc^1_{1,2}, pc^2_{1,1}), (pc^1_{1,1}, pc^2_{1,2}), (pc^1_{1,2}, pc^2_{1,2})\}$$
$$\text{PChoice}((tr^1_1, tr^2_2)) = \{(pc^1_{1,1}, pc^2_{2,1}), (pc^1_{1,2}, pc^2_{2,1})\}$$
$$\text{PChoice} = \text{PChoice}((tr^1_1, tr^2_2)) \cup \text{PChoice}((tr^1_1, tr^2_2))$$
$$\text{Assign}((tr^1_1, tr^2_1), (pc^1_{1,1}, pc^2_{1,2})) \equiv x' = 0 \wedge y' = x^2$$
$$p((tr^1_1, tr^2_1), (pc^1_{1,1}, pc^2_{1,2})) = 0.2 \cdot 0.6 = 0.12$$

Figure 3.1: Formalization of the parallel composition of two probabilistic hybrid automata $\mathscr{A}_1$ and $\mathscr{A}_2$ (guards are omitted for the sake of clarity). NChoice defines all possible combinations of local transitions and PChoice the corresponding combinations of transition variants. As an example the synchronous execution of transitions tr^1_1 and tr^2_1 with the variants $pc^1_{1,1}$ and $pc^2_{1,2}$ respectively is shown, which has the conjunctive predicate $x' = 0 \wedge y' = x^2$ as side effect. The combined probability of this execution is 0.12.

- A *probability distribution* $p(tr^i_j) \in P(PC_{tr^i_j})$, where $PC_{tr^i_j}$ is a finite and nonempty set of symbolic transition variants and $P(PC_{tr^i_j})$ denotes the set of probability distributions over $PC_{tr^i_j}$. Each transition tr^i_j has such a distribution $p(tr^i_j)$ providing $|PC_{tr^i_j}|$ many transition alternatives.
- For each transition alternative $pc \in PC_{tr^i_j}$ of transition tr^i_j an *assignment predicate* $\text{asgn}(tr^i_j, pc)$ defining the successor state. As transition guards, $\text{asgn}(tr^i_j, pc)$ is an arithmetic predicate in the arithmetic theory $\mathscr{T}$ over variables in $D_1, \ldots, D_n$ and $R_1, \ldots, R_n$ as well as primed variants thereof, the latter again representing the post-states.

Note that the guard and assignment predicates may again refer to the global pre- and post-state, i.e. the current states and the post-states of all concurrent automata in $\mathscr{S}$. This definition enables an automaton to read state variables of other automata, and moreover offers the possibility of non-local writes, entailing agreement in case of multiple concurrent updates to the same variables. Semantically, updates will only be performed in case all concurrent automata agree on them, and the system will become deadlocked in case of inconsistent updates. Furthermore, we require that the concurrent execution of assignments are deterministic wrt. the primed variables, i.e. the concurrent execution of the local transition alternatives of the individual automata uniquely determines the global post-state of the overall system.

The above two requirements imply that any concurrently enabled combination of local transitions may permit at most one successor state for each possible resolution of the local probabilistic choices. This condition necessitates a global view of transitions and their related assignments, which motivates the following definitions, as illustrated in Figure 3.1.

To obtain such a global view, let $\text{NChoice} = \prod_{i=1}^{n} Tr_i$ denote the Cartesian product of the local transition sets, thus representing the set of all potentially possible global transitions. As each local transition may have multiple probabilistic variants, the same applies for global transitions. With regards to a single global transition $(tr^1, \ldots, tr^n) \in \text{NChoice}$, the set of associated probabilistic transition alternatives is $\text{PChoice}((tr^1, \ldots, tr^n)) = \prod_{i=1}^{n} PC_{tr^i}$, which is the Cartesian product of the local probabilistic transition alternatives available for the individual local transitions $tr^1, \ldots, tr^n$. Taking together all the global probabilistic alternatives of all global transitions, $\text{PChoice} := \bigcup_{nc \in \text{NChoice}} \text{PChoice}(nc)$ denotes the set of all global probabilistic choices. Given a global non-deterministic transition choice $tr = (tr^1, \ldots, tr^n) \in \text{NChoice}$ and a corresponding global probabilistic alternative choice $pc = (pc^1, \ldots, pc^n) \in \text{PChoice}(tr)$, we denote by $\text{Assign}(tr, pc) = \bigwedge_{i=1}^{n} \text{asgn}(tr^i, pc^i)$ the conjunction of the selected local assignment predicates. With these definitions, we can formalize the requirements that each concurrent execution of local transition alternatives be deterministic: namely we require that for each global transition $tr \in \text{NChoice}$ and for each global probabilistic alternative $pc \in \text{PChoice}(tr)$, the associated global assignment $\text{Assign}(tr, pc)$ is deterministic or, equivalently, a partial function, i.e. it satisfies

$$\text{Assign}(tr, pc) \wedge \text{Assign}(tr, pc)[\vec{e}/\vec{d}', \vec{y}/\vec{x}'] \Rightarrow \vec{e} = \vec{d}' \wedge \vec{y} = \vec{x}',$$

where $\vec{d}'$ and $\vec{x}'$ denote the vectors of all primed discrete and continuous variables of all automata $\mathscr{A}_1, \ldots, \mathscr{A}_n$, respectively. Intuitively, the current global state and a global assignment uniquely determines the global post-state. Let $\text{States}_{\mathscr{S}} = \prod_{i=1}^{n} \text{States}_i$ be the global state space of system $\mathscr{S}$. In the sequel, we will define the concurrent semantics of the system $\mathscr{S}$. Here, all partners do propose a local transition which is fixed as soon as the partners have reached consensus, meaning the guards of the involved local transitions are consistent. This is done by checking whether a global post-state exists which together with the current pre-state satisfies the conjunction of the (generalized) local guards. Once the global transition has been negotiated, all partners do randomly select a local transition alternative. Provided that the assignments corresponding to the resulting global probabilistic alternative are consistent, each system enters the unique post-state of $\mathscr{S}$ arising due to determinacy of assignments. In case the selected global system step is impossible due to inconsistency between the selected guards of all $\mathscr{A}_i$ or due to inconsistency of the randomly selected assignments, the overall system $\mathscr{S}$ deadlocks in a distinguished state $\perp$.

Given a selection of transitions and transition alternatives, at most one post-state exists:

Property 1 (Existence and uniqueness of post-states)**.** *Let $\mathscr{S}$ be a system of concurrent discrete-time probabilistic hybrid automata. Further, let $s \in \text{States}_{\mathscr{S}}$ be a state of $\mathscr{S}$, $tr = (tr^1, \ldots, tr^n) \in \text{NChoice}$ be a non-deterministic transition choice, and $pc = (pc^1, \ldots, pc^n) \in \text{PChoice}(tr)$ be a probabilistic choice of transition alternatives. We define the predicate* $\text{val}(z)$ *for $z \in \text{States}_{\mathscr{S}}$ as a conjunction of equations $\bigwedge_{v \in \bigcup_{i=1}^{n}(D_i \cup R_i)} v = z(v)$, where $z(v)$ is the value of v in state z. Then, if*

$$\text{val}(s) \wedge \bigwedge_{i=1}^{n} \Big(g(tr^i) \wedge \text{asgn}(tr^i, pc^i) \Big)$$

is satisfiable then there exists exactly one state s' such that

$$\mathrm{val}(s) \wedge \mathrm{val}(s')' \wedge \bigwedge_{i=1}^{n} \Big(g(tr^i) \wedge \mathrm{asgn}(tr^i, pc^i) \Big)$$

is satisfiable, where $\mathrm{val}(.)'$ *is* $\mathrm{val}(.)$ *with all variable names decorated by primes.*

In this case, we denote by $\mathrm{Post}(s, tr, pc)$ *the unique post-state* s'*. Otherwise, the system deadlocks and we define* $\mathrm{Post}(s, tr, pc) = \bot$*. For convenience, we define* $\mathrm{Post}(\bot, tr, pc) = \bot$ *for all* tr, pc*.*

The semantics of $\mathscr{S}$ is then defined by *runs* of $\mathscr{S}$ that are finite alternating sequences of states and transitions, the latter involving both non-deterministic and probabilistic choices. A *run* $\langle s_0, (tr_1, pc_1), s_1, \ldots, (tr_k, pc_k), s_k \rangle$ *with* $s_0 \in \mathrm{States}_{\mathscr{S}}$*,* $s_i \in \mathrm{States}_{\mathscr{S}} \cup \{\bot\}$ *for* $i > 0$*, and* $(tr_i, pc_i) \in \mathrm{NChoice} \times \mathrm{PChoice}$ *of* $\mathscr{S}$ satisfies the following properties:

1. s_0 satisfies the initial predicate $\bigwedge_{i=1}^{n} init_i$,
2. $pc_j \in \mathrm{PChoice}(tr_j)$ for all $1 \leq j \leq k$.
3. $s_{j+1} = \mathrm{Post}(s_j, tr_j, pc_j)$ for all $0 \leq j \leq k-1$.

Thus, each run starts in the global initial state defined by the initial state predicates of the concurrent components. Upon each computation step, all concurrent automata first select non-deterministically among their transitions and then probabilistically under their variants. The corresponding transition step leads to a unique post-state, if existent, or to deadlock otherwise. The *probability of a transition step* from s to s' under non-deterministic choice $tr = (tr^1, \ldots, tr^n)$ and probabilistic choices $pc = (pc^1, \ldots, pc^n)$ is given by $p(tr, pc) = \prod_{i=1}^{n} p(tr^i)(pc^i)$ (cf. Figure 3.1). The deadlock state is instead considered to be absorbing. The *probability of a (finite) run* is the product of the probabilities of all transition steps of that run. The *length of a run* coincides to the number of transition steps involved. Note that the accumulated probability of all runs of a given length k under a given policy resolving non-determinism is always 1.

The presented definitions create a simple and sound way to cope with concurrent PHA exchanging information by shared state. For such models, the symbolic methods for analysis of step-bounded properties developed in the AVACS project [43, 44] generalize straightforwardly, permitting a symbolic encoding of size linear in the number of concurrent components [89], thus avoiding the state explosion incurred in product constructions. More complex concurrency models, like message passing, cannot currently be encoded as straightforwardly as the shared-state concurrency. In these models the individual components communicate indirectly using messages which are prone to (deterministic or random) message delay or which may even be lost with some probability.

4 Probabilistic Bisimulations

This Section provides a broad overview of the concept of probabilistic bisimulation applied to various classes of stochastic models. The concept of bisimulation, to be formalized shortly, induces an equivalence between two systems if they match each other's moves – in this sense, each of the systems cannot be distinguished from the other.

Section 4.1 focuses on discrete-space models, whereas Section 4.2 on general state-space models. Both Sections exclusively deal with discrete time models and provide references to literature on continuous-time models.

4.1 Notions for Discrete-Space Models

This Section focuses on discrete-space models, and mostly on the discrete-time case. The report will cover Probabilistic Automata, Interactive Markov chains, and DTMC.

4.1.1 Literature Overview

The concept of strong probabilistic bisimulation over a discrete-time, finite-state Markov chain has been introduced in [69], based on earlier notions for non-probabilistic models [72, 77]. The notion of weak bisimulation is discussed in [57], for a number of probabilistic processes. The contributions in [11, 42, 84] deal with more general notions, such as that of probabilistic simulation relations over respectively Markov chains, classes of probabilistic automata, as well as probabilistic hybrid automata.

With regards to Section 4.1.4, lumping and decomposition are well established methods for computing bisimulations on Markov chains. Focusing on performance analysis, Capra et al. [23] use lumping to reduce Markov chains. Derisavi contributed on the optimality of lumping [28, 29, 30, 31]. Besides lumping, system model decomposition [71] is a second focus in this section.

4.1.2 Bisimulation on Probabilistic Automata

Definition 20 (Bisimulation on PA)**.** *Let $(S, A, \rightarrow, s_0)$ be a probabilistic automaton and R an equivalence relation on S. Then, R is a* probabilistic bisimulation *on S if for any $(s, s') \in R$:*

$$\text{if } s \xrightarrow{\alpha} \mu \text{ then } s' \xrightarrow{\alpha} \mu' \text{ for some } \mu' \text{ and } \mu \equiv_R \mu',$$

where $\equiv_R$ is defined as follows: $\mu \equiv_R \mu'$ if and only if $\mu(C) = \mu'(C)$ for any equivalence class $C \subseteq S$ under R.

States s and s' are probabilistic bisimilar, denoted by $s \sim s'$, if there exists a probabilistic bisimulation R on S with $(s, s') \in R$.

Recall that a binary relation over a given set is an equivalence relation if it is reflexive, symmetric, and transitive.

Theorem 3. *$\sim$ is a congruence with respect to parallel composition and hiding.*

The above theorem thus asserts that whenever PA $\mathscr{P} \sim \mathscr{Q}$, then for any set H with $\tau \notin H$, we have that $\mathscr{P} \setminus H \sim \mathscr{Q} \setminus H$. In addition, for any $\mathscr{R}$ and set of actions A, $\mathscr{P} \|_A \mathscr{R} \sim \mathscr{Q} \|_A \mathscr{R}$. This result allows a component-based comparison of complex system models.

4.1.3 Bisimulation on Interactive Markov Chains

To compare IMCs, we introduce the notions of strong and weak bisimulation. For set $C \subseteq S$ and state s, let $\mathbf{R}(s,C) = \sum_{s' \in C} \mathbf{R}(s,s')$. Intuitively, two states s and t are strongly bisimilar if any interactive transition $s \xrightarrow{\alpha} s'$ can be mimicked by t, i.e., $t \xrightarrow{\alpha} t'$ such that s' and t' are bisimilar. In addition, the cumulative rate of moving from s to some equivalence class C of states, i.e., $\mathbf{R}(s,C)$ equals $\mathbf{R}(t,C)$. Since the probability of a Markovian transition to be executed immediately is zero, whereas internal interactive transitions take always place immediately, there is no need to require equality of cumulative rates if states have outgoing internal transitions. Let $s \xrightarrow{\tau}\!\!\!\!\!/\;\;$ denote a predicate that is true if and only if s has no outgoing τ-transition. For state s, action α and $C \subseteq S$, let $\mathbf{T}(s,\alpha,C) = 1$ if and only if $\{s' \in C \mid s \xrightarrow{\alpha} s'\}$ is non-empty.

Definition 21 (Strong bisimulation)**.** *Let $\mathscr{I} = \left(S, A, \longrightarrow, \Longrightarrow, s_0\right)$ be an IMC. An equivalence relation $R \subseteq S \times S$ is a* strong bisimulation *on $\mathscr{I}$ if for any $(s,t) \in R$ and equivalence class $C \in S/R$ (the quotient set) the following holds:*

1. *for any $\alpha \in A$, $\mathbf{T}(s,\alpha,C) = \mathbf{T}(t,\alpha,C)$, and*
2. *$s \xrightarrow{\tau}\!\!\!\!\!/\;\;$ implies $\mathbf{R}(s,C) = \mathbf{R}(t,C)$.*

States s and s' are strongly bisimilar, denoted $s \sim s'$, if $(s,s') \in R$ for some strong bisimulation R.

The rate equality is adopted from the notion of lumping equivalence [62]; in fact the above definition applied to CTMC exactly yields the notion of lumping. Two IMCs $\mathscr{I}_1$ and $\mathscr{I}_2$ on (disjoint) state spaces S_1 and S_2 respectively are bisimilar, denoted $\mathscr{I}_1 \sim \mathscr{I}_2$, if there exists a strong bisimulation R on $S_1 \cup S_2$ such that $(s_{0,1}, s_{0,2}) \in R$. The next property asserts that $\sim$ is substitutive with respect to parallel composition and hiding, so, e.g., $\mathscr{I} \sim \mathscr{I}'$ implies for any set A that $\mathscr{I} \setminus A \sim \mathscr{I}' \setminus A$.

Theorem 4. *[57] $\sim$ is a congruence with respect to parallel composition and hiding.*

As discussed before, τ-transitions play a special role in IMC. Whereas strong bisimulation treats all interactive transitions in the same way, regardless whether they are internal (i.e., labelled by τ) or not, weak bisimulation takes an observer's point of view and cannot distinguish between executing several successive τ-transitions or a single one. This allows for collapsing sequences of internal interactive transitions by a single such transition. This acts

exactly the same as for labeled transition systems. The treatment of Markovian transitions is a bit more involved, however. First, let us remark that the probability distribution of a sequence of exponential distributions is not an exponential distribution but constitutes a phase-type distribution. Therefore, it is not possible to define a weak version of the transition relation $\Longrightarrow$ as is done for weak bisimulation in labeled transition systems. The solution is to demand that Markovian transitions have to be mimicked in the strong sense, while they can be preceded and/or followed by arbitrary sequences of internal interactive transitions. The treatment of sequences of internal interactive transitions is similar to that of branching bisimulation [95]. As for strong bisimulation, rate equality is only required if a state has no outgoing internal transitions (maximal progress). Let $s \xrightarrow{\tau^*} s'$ denote that s' can be reached from s solely via zero or more τ-transitions; in particular $s \xrightarrow{\tau^*} s$ for any state s. For state s, action α and $C \subseteq S$, let $\mathbf{W}(s,\alpha,C) = 1$ if and only if $\{s' \in C \mid s \xrightarrow{\tau^*} \xrightarrow{\alpha} \xrightarrow{\tau^*} s'\}$ is non-empty. Let C^τ be the set of states that can reach some state in C via zero or more τ-transitions.

Definition 22 (Weak bisimulation)**.** *Let $\mathscr{I} = \left(S, A, \longrightarrow, \Longrightarrow, s_0\right)$ be an IMC. An equivalence relation $R \subseteq S \times S$ is a* weak bisimulation *on $\mathscr{I}$ if for any $(s,t) \in R$ and equivalence class $C \in S/R$, the following holds:*

1. *for any $\alpha \in A$, $\mathbf{W}(s,\alpha,C) = \mathbf{W}(t,\alpha,C)$, and*
2. *$s \xrightarrow{\tau^*} s'$ and $s' \not\xrightarrow{\tau}$ implies $t \xrightarrow{\tau^*} t'$ and $t' \not\xrightarrow{\tau}$ and $\mathbf{R}(s', C^\tau) = \mathbf{R}(t', C^\tau)$ for some $t' \in S$.*

States s and s' are weakly bisimilar, denoted $s \approx s'$, if $(s,s') \in R$ for some weak bisimulation R.

Theorem 5. *[57] $\approx$ is a congruence with respect to parallel composition and hiding.*

Assume that the IMC under consideration is complete, i.e., it is not subject any further to interaction with other components that are modeled as IMC. This is important, as this means that actions cannot be further delayed due to a delay which is imposed by the environment. Formally, this means that we can safely hide all actions in the IMC at hand, i.e., we consider $\mathscr{I} \setminus H$ where H contains all actions occuring in $\mathscr{I}$. Accordingly, all actions are labeled by τ. The typical specification that is subject to analysis is thus of the form:

$$\left(\mathscr{I}_1 \,||_{A_1}\, \mathscr{I}_2 \,||_{A_2} \ldots ||_{A_N}\, \mathscr{I}_N\right) \setminus H$$

where H is the union of all actions in IMC $\mathscr{I}_i$, i.e., $H = \cup_{i=1}^{N} A_i$. Due to the maximal progress assumption, the resulting IMC can be simplified: in any state that has a τ-transition, all Markovian transitions can be removed. Subsequently, sequences of τ-transitions can be collapsed by applying weak bisimulation.

If nondeterminism is absent in the resulting IMC, the model results in a CTMC, and all analysis techniques for CTMC can be employed such as transient and steady-state analysis, or CSL model checking [9] (Continuous Stochastic Logic, or CSL, is a probabilistic modal logic for classes of continuous-time stochastic processes).

4.1.4 Bisimulations based on Decomposition and Lumping

State-exploratory analysis of discrete-state models, be it Kripke structures in the qualitative setting or (discrete time) Markov Chains in the probabilistic setting, often suffers from the so-called state-space explosion. State-space explosion originates from the combinatorial explosion inherent to product constructions explaining the joint dynamics of concurrent components and thus confines our ability to analyze massively concurrent systems. In the following we present a combination of methods addressing this problem for certain concurrent compositions of Markov Chains.

Two techniques can be applied to deal with the complexity issues for such systems: *Decomposition* allows for piecewise analysis and *lumping* allows for reduction of the state space (and thereby of the transition space). When combined, *piecewise* reduction can be carried out on the sub-systems local state spaces. The recomposition of the reduced subsystems is an *exact* bisimulation to the original Markov chain [73]. Applying lumping on individual subsystems circumvents the need to analyze the whole system at once, which is likely to be intractable. Instead, with a "divide and conquer" strategy, the class of systems that are tractable to analyze is expanded. In case a model remains intractable for analysis even after (piecewise) reduction, approximate methods presented in the following offer a solution for the price of accuracy. The bisimilar character of the "divide and conquer" strategy is shown for discrete-time Markov chains.

Discrete System Model

The state space S and transition probabilities $\mathbf{P}$ of a DTMC $\mathscr{D} = (S, \mathbf{P}, L, \mu_0)$ are defined by a distributed system. A distributed system comprises a finite number of processes. Each process has one register to store data from a finite domain. As a consequence, the DTMC for the distributed system has a finite number of states (and thereby transitions). Probabilistic attributes of the distributed system like execution semantics, (probabilistic) scheduler, algorithm and fault model specify the transition probabilities between the states.

The divide and conquer approach splits a system into its smaller sub-system components. Thereafter, the Markov chain of each subsystem is computed. Lumping (i.e. the aggregation of states to prune redundant information) can be applied to the sub-Markov chains using a given equivalence class. When the reduced sub-Markov chains are composed according to the prior decomposition, the result is exactly bisimilar to the original Markov chain, yet likely to be considerably smaller and thereby more tractable for analysis.

Consider a distributed system comprising n processes $\pi_1, \ldots \pi_n$ connected via bidirectional communication channels. Each two connected processes are called neighbors. Each process has a memory register r_i and reads from the registers of its neighbors and writes to its own register. Thus, the system state s_t at time t can be characterized by $s_t = \langle r_1, \ldots, r_n \rangle$.

A canonical simplification abstracts the state space with a three value-based logic. When a register stores the intended value, the abstraction labels the register with a zero: $r_i = 0$. It stores $r_i = 1$ when π_i is (knowingly) unable to compute the correct value, and $r_i = 2$ when the register contains a wrong value. A system satisfies the global safety predicate $c_t \models \mathscr{P} : \forall r_i : r_i = 0$ when all registers store a 0.

Processes cannot distinguish between correct and incorrect information provided by their neighbors *locally*. They can detect the presence of faults when they are provided with ambiguous information. Then, they set their register to $r_i = 1$. Otherwise, they copy the information from their neighbors. With the three value-based logic the state space contains up to 3^n states.

The processes execute the distributed algorithm introduced in [74, p.24] and recapitulated as follows. Process π_1 is selected as distinguished *root* process, as required by the algorithm. It stores the correct value when no fault occurs $r_1 = 0$ and an incorrect value otherwise $r_1 = 2$. The other (non-root) processes copy the values provided by those neighbors that are closer to the root and store the correct value $r_i = 0$ when they read the correct and not the incorrect value. They store the incorrect value $r_i = 2$ when they read an incorrect value and not a correct value. They store the *fail safe* value otherwise $r_i = 1$ (the abstraction does not distinguish between incorrect values).

Assume that faults only corrupt the registers of the executing processes. Consider further a probabilistic scheduler and serial execution semantics such that in each computation step exactly one process is randomly selected to execute with probability $e = \frac{1}{n}$. With probability p the executing process executes as desired and with probability $q = 1 - p$ the register of the executing process is corrupted by a fault. Serial execution semantics demand a different composition technique, opposed to the concurrent composition shown in Section 3.1.

The system is split into subsets $X_1, \ldots, X_m$ such that each subset is a connected component of the process graph. Slicing systems in order to apply lumping to the sub-systems Markov chains (i.e. system decomposition with minimal cuts and maximal tractable size of subsystems) is discussed in [73]. We label

- the sub-Markov chain that models subsystem X_i with $\mathscr{D}_i$,
- the Markov chain $\mathscr{D}_i$ excluding shared processes with *underlying* (or sibling) subsystems $\mathscr{D}_{i,-}$ (the order between sub-systems defines their order within the system),
- and Markov chains that model the behavior of an intersecting process π_i (that connects two or more sub-systems) with $\mathscr{D}_{\pi_i}$.

In a "divide and conquer" approach we decompose the system, apply lumping on the subsystems, and recompose the lumped subsystems to possibly arrive at a considerably smaller Markov chain that is a bisimulation (regarding the equivalence relation $\sim$ presented in the following section) to the DTMC of the system.

Decomposition

The decomposition comprises three main steps: decomposition, lumping and recomposition. The three steps are presented briefly, outlining the basic idea behind the approach. A detailed report is in [73].

Step 1. The initial subsystem X_1 with $\pi_1 \in X_1$ is transformed into a (sub-)DTMC $\mathscr{D}_1$, obtained from $\mathscr{D}$, and consequently split into transients $\mathscr{D}_{\pi_i}$ and residents $\mathscr{D}_{1,-}$. Transients are

those processes connecting two subsystems while residents are processes that only belong to one subsystem. The first step is to compute $\mathscr{D}_1$, the DTMC modeling the behavior of X_1. The relation of the scheduler election probabilities e between X_1 and the rest of the system play an important role. As only a subset of X is considered during the decomposition, chances are that either one process in X_1 executes, or a process outside X_1 executes. DTMC $\mathscr{D}_1$ must take the possibility into account that a process outside X_1 executes. In that case, X_1 remains in its state for one step.

To account for these global scheduling relations between the subsystems, all transition probabilities in $\mathscr{D}_1$ are multiplied with $p(X_1)$, the chance that a process in X_1 executes. The diagonal elements of the transition probability matrix $\mathscr{D}_1$ are the probabilities that the sub-system remains in its state. To these diagonal elements the probability $1 - p(X_1)$ (the probability that a process outside X_1 executes) is added. Thereby, the scheduler selection probabilities are taken into account.

When $\mathscr{D} \neq \mathscr{D}_1$ (i.e. X was split into more than one part), then X_1 and its neighbors (the subsystems to which X_1 propagates faults directly) share at least one mutual process (transient). When computing the sub-Markov chains in which $\mathscr{D}_1$ propagates (unidirectional propagation from root towards leafs as proposed in [73]), they take the transient(s) as their local root(s). The intersecting process (i.e. the transient) is excluded from $\mathscr{D}_1$ which is thereby split into $\mathscr{D}_{1,-}$ (the DTMC that considers the residents only) and a Markov chain $\mathscr{D}_{\pi_i}$ for each transient.

For the decomposition (with a focus on the transient processes) three rules are important:

1. A sub-system can cut out one or more transients that become the local root process for other subsystems.

2. A transient can become the root for one or more sub-systems but must remain in only one sub-system.

3. A sub-system can have multiple (local) root processes.

Then, the sub-Markov chains of those systems into which X_1 propagates are computed. The double index in $\mathscr{D}_{1,-}$ refers to the original Markov chain $\mathscr{D}_1$ and the "$-$" refers to the fact that all transients were cut out. The result is $\mathscr{D}_1 = \mathscr{D}_{1,-} \otimes \mathscr{D}_{\pi_i}$ (analogously for multiple transients). The $\otimes$ operator is the *serial* composition of two Markov chains. In contrast to the concurrent composition (Cartesian product), serial execution semantics prohibit concurrency which must be taken into account during the recomposition.

Step 2. A subsystem $X_i, i > 1$, shares one or more processes with subsystems that are *closer to* X_1 than itself (i.e. the neighboring subsystems that have a shorter distance to the root process than X_1 itself). The Markov chains describing the behavior of these processes can be computed in a recursive manner. Once characterized, $\mathscr{D}_i$ is computed in the same way as described for $\mathscr{D}_1$. Common processes with underlying subsystems (i.e. subsystems that are farther away from X_1 than X_i itself) can be split as described above. A single process can be part of more than two subsystems.

Step 3. Eventually, the lowest sub-systems (i.e. subsystems that do not propagate in other subsystems) are reached. They take the input from those neighboring their superior subsystems.

Remark 2. *While the Cartesian product (concurrent composition) of both sub-Markov chains $\mathscr{D}_1 = \mathscr{D}_{1,-} \times \mathscr{D}_{\pi_i}$ is applicable for parallel execution semantics (where multiple processes are allowed to change their individual register every time step), serial execution semantics demand serial composition. The serial composition does not lead to transitions in which more than one register changes its value per time step during the re-composition.*

Reduction

The lumping of general DTMCs is described in [74] and also applicable on sub-DTMCs. A sub-Markov chain comprises states and transition probabilities between these states. The conditions that qualify two states for lumping have been discussed in [75] and are formally defined in Definition 23. The reduced version of $\mathscr{D}_i$ is labeled with $\mathbb{D}_i$.

Recomposition

To recompose subsystems, we take the relevant (possibly reduced) sub-Markov chains and combine them using the $\otimes$ operator for serial composition. We label the recomposed Markov chain $\mathbb{D}$. If we skipped the reduction (lumping), the result is the original Markov chain.

Formal Method and Proof

The Markov chain $\mathscr{D}$ that models the behavior of the system comprises states S that are connected via transition probabilities $\mathbf{P}$. We label the initial probability distribution over S with μ_0, that after m iterations with μ_m, and the stationary distribution with μ_∞ (and respectively μ_0^{red}, μ_n^{red} and μ_∞^{red} for the *reduced* DTMC $\mathbb{D}$).

To analyze the effect of lumping on a specific safety predicate, we define the following equivalence relation. Two states s_i and s_j belong to the same equivalence class if they either both satisfy or both dissatisfy the safety predicate $\mathscr{P}$ and have equal transition probabilities for each of their target states as defined in Definition 23. We label the equivalence class of s under $\sim$ with $[s]_\sim$.

Definition 23. *Let us introduce the following equivalence relation:*

$$\begin{aligned} s_i \sim s_j :\Leftrightarrow \quad & ((s_i \models \mathscr{P} \vee s_j \models \mathscr{P}) \wedge \\ & (s_i \not\models \mathscr{P} \vee s_j \not\models \mathscr{P})) \vee \\ & \forall s \in S : p(s_i \mid s) = p(s_j \mid s) \end{aligned}$$

We reduce $\mathscr{D}$ with $red(\mathscr{D}, \mathscr{P})$ and fit the safety predicate via the following definition:

Definition 24.

$$red(\mathscr{D}, \mathscr{P}) = (\mathbb{D}, \mathbb{P}) \tag{4.1}$$

$$\begin{array}{ccc} (\mathscr{D}, \mathbf{P}_0) & \xrightarrow{[\sim]} & (\mathbb{D}, \mathbf{P}_0^{red}) \\ \downarrow & & \downarrow \\ (\mathscr{D}, \mathbf{P}_\infty) & \xrightarrow{[\sim]} & (\mathbb{D}, \mathbf{P}_\infty^{red}) \end{array}$$

Figure 4.1: Commutative Diagram for Equivalence of Stationary State Probability Distribution via Equivalence Class $[]_\sim$

$$\mathbb{D} = (S_{red}, \mathbf{P}_{red}) \tag{4.2}$$

$$S_{red} = \{[s]_\sim | s \in S\} \tag{4.3}$$

$$\mathbf{P}_{red}([s_i]_\sim, [s_j]_\sim) = \sum_{d_i \in [s_i]_\sim} \mathbf{P}(d_i, d_j), d_j \in [s_j]_\sim \tag{4.4}$$

$$[s]_\sim \models \mathbb{P} :\Leftrightarrow \exists d \in [s]_\sim : d \models \mathscr{P} \tag{4.5}$$

Remark 3. *In equation 4.4 choice of d is arbitrary because of the equalence established in Definition 23.*

Equation (4.3) describes the *state lumping* and equation 4.4 the *transition lumping*. The reduction by Definition 24 lumps those states that are in the same equivalence class $[s]_\sim$ (transitions $\mathbf{P}$ respectively). The constraints that qualify states for equivalence classes are defined in Definition 23.

The safety predicate $\mathscr{P}$ is defined for the state space S of $\mathscr{D}$. We require an (analogously lumped) predicate $\mathbb{P}$ that matches the reduced state space of $\mathbb{D}$, shown in Equation 4.5, which describes the *predicate lumping*. In order to show the bisimulation between $\mathscr{D}$ and $\mathbb{D}$, it is necessary to show that both have an equal steady state probability distribution, which is done next.

Theorem 6. *The following holds:* $\mu_\infty^{red}([s]_\sim) = \sum_{d \in [s]_\sim} \mu_\infty(d)$.

Proof. We show by induction that both the original and the reduced Markov chains result in the same stationary probability distribution regarding the equivalence classes, as shown in Figure 4.1.

Let μ_0 be an arbitrary initial distribution for $\mathscr{D}$ and let $\mu_0^{red}([s]_\sim) = \sum_{d \in [s]_\sim} \mu_0(d)$ be the corresponding initial distribution for $\mathbb{D}$. We show that for μ_m and μ_m^{red}, which are the probability distributions for $\mathscr{D}$ and $\mathbb{D}$ at time point m with an initial distribution μ_0 and μ_0^{red}, the following holds:

$$\forall m : \mu_m^{red}([s]_\sim) = \sum_{d \in [s]_\sim} s_m(d). \tag{4.6}$$

Proof via induction over m.
Anchor: $m = 0$ holds by assumption.

Step: the following holds

$$\mu^{red}_{m+1}([s]_\sim) = \sum_{[d]_\sim \in S^{red}} \mu^{red}_m([d]_\sim) \cdot \mathbf{P}^{red}([d]_\sim, [s]_\sim) \tag{4.7}$$

$$= \sum_{[d]_\sim \in S^{red}} \Big(\sum_{e \in [d]_\sim} \mu_m(e) \Big) \cdot \Big(\sum_{f \in [s]_\sim} \mathbf{P}(d, f) \Big) \tag{4.8}$$

$$= \sum_{[d]_\sim \in S^{red}} \sum_{e \in [d]_\sim} \sum_{f \in [s]_\sim} \mu_m(e) \cdot \mathbf{P}(d, f), \tag{4.9}$$

and with $\mathbf{P}(e, f) = \mathbf{P}(d, f)$ because of $e \sim d$ (cf. Definition 23)

$$= \sum_{[d]_\sim \in S_{red}} \sum_{e \in [d]_\sim} \sum_{f \in [s]_\sim} \mu_m(e) \cdot \mathbf{P}(e, f) \tag{4.10}$$

$$= \sum_{e \in S} \sum_{f \in [s]_\sim} \mu_m(e) \cdot \mathbf{P}(e, f) \tag{4.11}$$

$$= \sum_{f \in [s]_\sim} \sum_{e \in S} \mu_m(e) \cdot \mathbf{P}(e, f) \tag{4.12}$$

$$= \sum_{d \in [s]_\sim} \mu_{m+1}(d). \tag{4.13}$$

Thus, $\forall m : \mu^{red}_m([c]_\sim) = \sum_{d \in [s]_\sim} \mu_n(d)$, which proves the Theorem. □

Corollary 1. *Theorem 6 and the first two conditions from Definition 23 allow to conclude that that the sum of the probability masses over all states satisfying the safety predicate is equal for each time step.*

The proof that both Markov chains $\mathscr{D}$ and $\mathbb{D}$ are exactly bisimilar with respect to the equivalence class $[]_\sim$ as shown in [73] is done analogously.

4.2 Notions for Continuous-Space Models

This Section focuses on general state-space processes. The main message is that for this class of processes an approximate version of the notion of probabilistic bisimulation is desirable. The survey thus shifts the focus to the development of approximate definitions.

4.2.1 Literature Overview

The use of approximate notions of bisimulations is advocated in [49] and motivated by robustness issues related to the verification of specifications over probabilistic models. The work in [35] discusses approximate notions of bisimulations for finite state labeled Markov chains, and elaborates on this notions by using an approach based on logical, as well as one based on games. Approximate notions appear much less restrictive than exact ones, particularly when applied over models with continuous state spaces – this is precisely what

has been observed also for deterministic models, where notions of exact bisimulation have been developed only for limited classes of models, e.g. timed automata [7], linear hybrid automata [56], o-minimal hybrid systems [68]. The introduction of approximate versions [50] based on distance between trajectories of deterministic models has lead to the study of approximate abstractions for nonlinear [79] and switched systems [51].

For continuous space processes (namely, discrete-time labeled Markov processes), [32] provides a relational and logical characterization of bisimulation. Alternatively, probabilistic bisimulations relations can be introduced via coalgebraic [27] or categorical arguments [92]. Building on these results, the material in [33] discusses metrics for labeled Markov processes, whereas [34] proposes metrics via weak bisimulations.

Related to the notions above, [88] introduces exact bisimulations for communicating piecewise-deterministic Markov processes, [36] discusses bisimulation of continuous-time processes, whereas [20] attempts definitions of bisimulations for stochastic hybrid models [16, 24]. None of these works proposes approximate variants of the the corresponding exact notions.

With focus on probabilistic models and on the development of metrics over systems realizations [50], the concept of probabilistic bisimulation functions is introduced in [61] and elaborated in [1]. The recent work in [91] puts forward a reachability problem to find metrics between discrete-time stochastic processes.

From a different perspective, [3] puts forward an approach based on randomization techniques to characterize approximation distances between processes over finite time horizons, with no assumptions on their dynamics. Along this line of research, [37] introduces an approximation for such processes, which can be related to the work in [90] (which uses Wasserstein Pseudometrics over continuous space processes) and to the classical reference in [66] (which discusses weak approximations of stochastic processes).

4.2.2 Characterization

Next, we provide a few details on the notion of exact and approximate bisimulation. We focus on two different perspectives — the first is through the definition of metrics over probability measures, whereas the second is process-based and looks at metrics between realizations.

Let us consider discrete time controlled Markov processes $\mathfrak{S}$ as defined in Section 2.2. However, to keep notation light, we shall simply refer to $\mathfrak{S} = (\mathscr{S}, \mathscr{T}, \mathscr{U})$, where $\mathscr{S}$ is an abstract state space (which is continuous, possibly hybrid), $\mathscr{T}$ is the transition kernel (corresponding to T_s for DTSHS), and $\mathscr{U}$ is an abstract control space (possibly, made up of reset and transition inputs).

Let us remark that $\mathscr{S}$ is a topological space that is homeomorphic to a subset of a complete and separable metric space. The reference metric can be taken to be equivalent to the usual Euclidean one. Furthermore, we assume that the space is endowed with a Borel σ-algebra.

Again $\mathscr{T}$ is the conditional stochastic kernel that assigns to each point $s \in \mathscr{S}$ and control $u \in \mathscr{U}$ a probability measure $\mathscr{T}(\cdot|s,u)$. For any set $A \in \mathscr{B}(\mathscr{S}), Prob_{s,u}(A) = \int_A \mathscr{T}(ds|s,u)$, where $Prob_{s,u}$ denotes the conditional probability $Prob(\cdot|s,u)$. Process $\mathfrak{S} = (\mathscr{S}, \mathscr{T}, \mathscr{U})$ is initialized according to a probability distribution $\pi : \mathscr{B}(\mathscr{S}) \to [0,1]$.

The syntax of the model above corresponds to the following semantics for a trajectory $\mathbf{s}(k)$ over the time horizon $[0,N]$. Let us fix a control string $\{u_0,u_1,\ldots,u_{N-1},u_i \in \mathscr{U}\}$. Given an initial condition $x \in \mathscr{S}$ sampled from the probability distribution π, and given the control input $u_0 \in \mathscr{U}$, the value of the process at time $k=1$, namely s_1, is described by a probability law characterized by the conditional kernel $\mathscr{T}(\cdot|x,u_0)$. Likewise, for any $k=1,\ldots,N-1, s_{k+1} \sim \mathscr{T}(\cdot|s_k,u_k)$.

Definition 25 (Exact Probabilistic Bisimulation)**.** *Consider two systems $\mathfrak{S}_1 = (\mathscr{S}_1,\mathscr{T}_1,\mathscr{U}_1)$ and $\mathfrak{S}_2 = (\mathscr{S}_2,\mathscr{T}_2,\mathscr{U}_2)$. A relation $R \subseteq \mathscr{S}_1 \times \mathscr{S}_2$ is a simulation relation of $\mathscr{S}_1$ by $\mathscr{S}_2$ if, whenever $s\,R\,t$ for any $s \in \mathscr{S}_1, t \in \mathscr{S}_2$, for any $u_1 \in \mathscr{U}_1$ and set $\tilde{S} \in (\mathscr{S}_1 \times \mathscr{S}_2)/R$ (which is Borel measurable), there exists a $u_2 \in \mathscr{U}_2$ such that*

$$\mathscr{T}_1(\tilde{S}|_{\mathscr{S}_1}|s,u_1) = \mathscr{T}_2(\tilde{S}|_{\mathscr{S}_2}|t,u_2),$$

where $\tilde{S}|_{\mathscr{S}_i}$ denotes the projection of $\tilde{S}$ on $\mathscr{S}_i$.
If R is also a simulation of $\mathscr{S}_2$ by $\mathscr{S}_1$, then R is an equivalence relation that is a bisimulation on $\mathscr{S}_1 \times \mathscr{S}_2$. A pair of states $s \in \mathscr{S}_1, t \in \mathscr{S}_2$ is said to be (probabilistically) bisimilar if $\exists R$, an equivalence relation, such that $s\,R\,t$, whereas two models $\mathfrak{S}_1, \mathfrak{S}_2$ are said to be (probabilistically) bisimilar (denoted $\mathfrak{S}_1\,R\,\mathfrak{S}_2$) if there exists a bisimulation relation (R) for any pair of states in respectively $\mathscr{S}_1$ and $\mathscr{S}_2$. □

Based on this definition, a logic $\mathscr{L}$ can be defined that allows to show that two states are bisimilar if and only if they satisfy the same formulas ϕ of the logic $\mathscr{L}$ [32].

The exact relational and logical characterizations are formal, but as discussed should be relaxed to accommodate for robustness and for real-world engineering applications. This leads to the notion of approximate bisimulation with level ϵ, or simply of ϵ-bisimulation [35]. Let R be a relation on a set A. A set $\tilde{A} \subseteq A$ is said to be R-closed if $R(\tilde{A}) = \{t|s\,R\,t, s \in \tilde{A}\} \subseteq \tilde{A}$.

Definition 26 (Approximate Probabilistic Bisimulation)**.** *Consider two models $\mathfrak{S}_1 = (\mathscr{S}_1,\mathscr{T}_1,\mathscr{U}_1)$ and $\mathfrak{S}_2 = (\mathscr{S}_2,\mathscr{T}_2,\mathscr{U}_2)$. A relation $R_\epsilon \subseteq \mathscr{S}_1 \times \mathscr{S}_2$ is an ϵ-bisimulation relation if, whenever $s\,R_\epsilon\,t$ for any $s \in \mathscr{S}_1, t \in \mathscr{S}_2$, and for any $u_1 \in \mathscr{U}_1$ and R_ϵ-closed set $\tilde{S} \subseteq \mathscr{S}_1 \times \mathscr{S}_2$, there exists a $u_2 \in \mathscr{U}_2$ (and similarly, for any $u_2 \in \mathscr{U}_2$ and R_ϵ-closed set $\tilde{S} \subseteq \mathscr{S}_1 \times \mathscr{S}_2$, there exists a $u_1 \in \mathscr{U}_1$) such that*

$$\left|\mathscr{T}_1(\tilde{S}|_{\mathscr{S}_1}|s,u_1) - \mathscr{T}_2(\tilde{S}|_{\mathscr{S}_2}|t,u_2)\right| \leq \epsilon,$$

where $\tilde{S}|_{\mathscr{S}_i}$ denotes the projection of $\tilde{S}$ on $\mathscr{S}_i$. In this case we say that the two systems are ϵ-bisimilar (denoted $\mathfrak{S}_1\,R_\epsilon\,\mathfrak{S}_2$). □

As discussed, a logic $\mathscr{L}$ can be defined that allows to show that two states are bisimilar if and only if they satisfy the same formulas ϕ of the logic $\mathscr{L}$. Similarly, probabilistic bisimulation can be sufficiently characterized by a family of functional expressions [33]. Given a process $\mathfrak{S}$, consider a family $\mathscr{F}^c$ of real-valued functions $f_{\mathfrak{S}} : \mathscr{S} \to [0,1]$, which are defined by a grammar, namely a set of operations that can be related to the rules of the logic

$\mathscr{L}$. Consider two processes $\mathfrak{S}_i = (\mathscr{S}_i, \mathscr{T}_i, \mathscr{U}), i = 1, 2$. A family $\mathscr{F}^c$ of functional expressions on $\mathfrak{S}_i$ induces a distance as follows:

$$d^c(\mathfrak{S}_1, \mathfrak{S}_2) = \sup_{f \in \mathscr{F}^c} |f_{\mathfrak{S}_1} - f_{\mathfrak{S}_2}|,$$

where $f_{\mathfrak{S}_i}$ are functions in $\mathscr{F}^c$ evaluated over the respective spaces $\mathscr{S}_i$. It can be shown that, for any $c \in (0, 1]$, d^c is a pseudo-metric. The use of a metric between processes allows to relate the distance in time between processes that are "similar" – in particular, approximately bisimilar. More precisely, it can be shown that if two processes $\mathfrak{S}_1, \mathfrak{S}_2$ are approximately bisimilar, then their distance $d^c(\mathfrak{S}_1, \mathfrak{S}_2)$ has a finite, explicit upper bound.

We now change gear and look at a second procedure, which investigates distance metrics between trajectories of the two processes. The second procedure exploits the dynamics of the two processes to define such metrics. Consider a model $\mathfrak{S}_1 = (\mathscr{S}_1, \mathscr{T}_1, \mathscr{U}_1)$ with associated realizations $\mathbf{s}_1(k), k \in \mathbb{N}$, and similarly a second model $\mathfrak{S}_2$. The quantification of similarity between $\mathfrak{S}_1$ and $\mathfrak{S}_2$ can be assessed by comparing trajectories of the two models. A formal comparison can be set up by seeking a function $g : \mathscr{S}_1 \times \mathscr{S}_2 \to \mathbb{R}_0^+$ that induces a metric over the distance between the trajectories [61]. If $\mathscr{S}_1 \neq \mathscr{S}_2$, in order to effectively relate the two processes, we need to assume the existence of output maps $\mathscr{Y}_i : \mathscr{S}_i \to \mathscr{S}^o$ taking values over the same observation space $\mathscr{S}^o$. In this instance, we would consider a function

$$g(x_1, x_2) = \|\mathscr{Y}_1(x_1) - \mathscr{Y}_2(x_2)\|^2.$$

If $\mathscr{S}_1 = \mathscr{S}_2$, then we can simply select $g(x_1, x_2) = \|x_1 - x_2\|^2$.

Given such a measurable, non-negative function g evaluated over $(\mathbf{s}_1(k), \mathbf{s}_2(k)), k \in \mathbb{N}$ (the Markov process related to the joint system $(\mathfrak{S}^1, \mathfrak{S}^2)$), the quality of the approximation between $\mathfrak{S}_1$ and $\mathfrak{S}_2$ is then characterized by the following quantity:

$$V_\delta^N(x) = Prob_x \left\{ \sup_{0 \leq k \leq N} g(\mathbf{s}_1(k), \mathbf{s}_2(k)) \geq \delta \right\}, \tag{4.14}$$

where $x \in \mathscr{S}_1 \times \mathscr{S}_2$ is a pair of initial conditions, $N \in \mathbb{N} \cup \{\infty\}$ denotes the time horizon, and δ is a non-negative real number denoting the approximation quality.

It is of interest to provide meaningful and possibly tight bounds for the probabilistic quantity in (4.14). Let us recall the following classical notion [38]:

Definition 27 ((Super-) Martingale). *Consider an autonomous stochastic process $\mathbf{x}(k), k \geq 0$, taking values in $\mathscr{S}$. A function $\chi : \mathscr{S} \to \mathbb{R}$ is called a martingale for the process $\mathbf{x}(k), k \geq 0$ if for any $x = \mathbf{x}(0) \in \mathscr{S}, k \geq 0, \mathbb{E}_x[\chi(\mathbf{x}(k))] = \chi(x)$. The function χ is called a* supermartingale *if $\mathbb{E}_x[\chi(\mathbf{x}(k))] \leq \chi(x)$.* □

Let us introduce the notion of stochastic bisimulation function (SBF), as introduced for continuous-time models in [61].

Definition 28 (Stochastic Bisimulation Function). *Let the measurable function $\varphi : \mathscr{S}_1 \times \mathscr{S}_2 \to \mathbb{R}_0^+$ satisfy the following conditions*

1. $\varphi(x) \geq g(x)$ *for all* $x \in \mathscr{S}_1 \times \mathscr{S}_2$*;*
2. *for any* $u_1 \in \mathscr{U}_1$*, there exists* $u_2 \in \mathscr{U}_2$ *such that the function* $\left(\varphi(\mathbf{x}_1(k), \mathbf{x}_2(k))\right)_{k\geq 0}$ *is a supermartingale for any fixed* $x \in \mathscr{S}_1 \times \mathscr{S}_2$*.*

Then φ *is a stochastic simulation function of* $\mathfrak{S}_1$ *by* $\mathfrak{S}_2$*. If* φ *is also a stochastic simulation function of* $\mathfrak{S}_2$ *by* $\mathfrak{S}_1$*, then it is an SBF for the function* g *with respect to the joint process* $(\mathfrak{S}_1, \mathfrak{S}_2)$*. If two processes admit an SBF, they are said to be probabilistically bisimilar (with precision* $\varphi(x)$*).* □

The existence of an SBF can be directly used to compute an upper bound for the quantity in (4.14). More precisely, selecting a parameter $\delta > 0$, any two initial conditions $x_i \in \mathscr{S}_i, i = 1, 2$, and by resorting to the properties of the SBF (as described in Definition 28) and to the Markov inequality [38], the following holds:

$$\begin{aligned} &Prob_{(x_1,x_2)}\left(\sup_{0\leq k<\infty} \|\mathscr{Y}_1(\mathbf{x}_1(k)) - \mathscr{Y}_2(\mathbf{x}_2(k))\|^2 \geq \delta\right) \\ &= Prob_{(x_1,x_2)}\left(\sup_{0\leq k<\infty} g\left(\mathbf{x}_1(k), \mathbf{x}_2(k)\right) \geq \delta\right) \\ &\leq Prob_{(x_1,x_2)}\left(\sup_{0\leq k<\infty} \varphi\left(\mathbf{x}_1(k), \mathbf{x}_2(k)\right) \geq \delta\right) \\ &\leq \frac{\varphi(x_1,x_2)}{\delta}. \end{aligned} \tag{4.15}$$

The knowledge of an SBF allows deriving bounds on the approximation quality between two processes. Next, we survey two conceptually different approaches to find such an SBF.

The contribution in [61] puts forward conditions to construct an SBF for certain classes of continuous-time stochastic processes, namely models that are linear in the drift, in the diffusion coefficient, and in the observation map. The setup allows for spontaneous jumps (with homogeneous arrivals) with related (linear) resets, thus resulting in a model with hybrid structure. The contribution raises structural assumptions on the joint process under study to derive sufficient conditions for the existence of an SBF with a particular form. These conditions can be shown to lead to certain stochastic stability properties of the models under study [41].

The contribution in [1] introduces sufficient conditions for the existence of an SBF, based on the use of contractivity analysis [70] for probabilistic systems. Furthermore, it shows that the notion of stochastic contractivity is related to a probabilistic version of the concept of incremental stability. In contrast to the first approach, the contractivity conditions are directly computable on the system dynamics; also, the probabilistic bisimulation function is directly obtained; finally, the conditions are applicable to nonlinear dynamics; however, at present the former results have applicability to hybrid models with richer dynamics. Both approaches can potentially yield bounds that are conservative.

The cadre of definitions and concepts described above allow establishing metrics for quantifying a-priori the similarity between the trajectories of two processes considered over an infinite time horizon. As such, they relied on structural assumptions over the models under study.

Alternatively, an approach described in [3] has the advantage to be valid for general models. It examines trajectories of the two processes over finite horizons. In other words, while the approaches above focused on the syntax of the models, this technique directly exploits the process semantics.

Consider two autonomous processes $\mathfrak{S}_1, \mathfrak{S}_2$, for which equation (4.15) can be re-written as follows:

$$Prob_x\big(d_T(\mathfrak{S}_1, \mathfrak{S}_2) > \gamma\big) \leq \epsilon \quad \Leftrightarrow \quad Prob_x\big(d_T(\mathfrak{S}_1, \mathfrak{S}_2) \leq \gamma\big) \geq 1 - \epsilon,$$

where $d_T(\cdot, \cdot)$ represents a metric between trajectories evaluated over the finite time horizon $[0, T]$ and started at $x \in \mathscr{S}_1 \times \mathscr{S}_2$, whereas γ is a given desired parameter quantifying the approximation precision, and ϵ is an a-priori quantity characterizing the similarity of the two processes of the models.

The quality γ of the approximation up to level $1 - \epsilon$ can be assessed as the solution of the following chance-constrained optimization problem [21]:

$$\min_{\gamma \in \Re} \gamma, \quad \text{subject to:} \qquad Prob_x\big(d_T(\mathfrak{S}_1, \mathfrak{S}_2) \leq \gamma\big) \geq 1 - \epsilon. \tag{4.16}$$

Denote with γ_ϵ the solution of (4.16): while its computation of this solution is often hard, it can be mitigated by using a randomized approach, which provides an estimate of γ_ϵ with approximation guarantees.

The randomized algorithm extracts N trajectories of the two processes $\mathfrak{S}_1, \mathfrak{S}_2$ over $[0, T]$, for random extractions of the initial condition x and of the driving uncertainty. It then computes their distance $d_T(\mathfrak{S}_1, \mathfrak{S}_2)$ and discards the $k < N$ obtained largest values, thus finding an approximate solution $\hat{\gamma}_\epsilon$. Based on arguments developed in [22], the work in [3] shows that a proper choice of the parameters k, N allows ensuring the feasibility of the solution $\hat{\gamma}_\epsilon$ (namely, the verification of the probabilistic constraint), and provides bounds on its performance degradation. Intuitively, by extracting at random N executions of the processes and discarding a-posteriori a fraction k/N of them that corresponds to the largest discrepancies between the processes, one can improve the quality bound γ while guaranteeing that the violation set has size smaller than or equal to the prescribed ϵ value.

The main advantage of this approach over those based on the synthesis of a probabilistic bisimulation function is the absence of assumptions on the dynamics of the two processes. On the other hand, the limitations are the presence of a confidence level on the obtained bounds; the validity of the outcomes over finite time horizons, and as of yet, the absence of an approach for non-autonomous models.

Bibliography

[1] A. Abate. A contractivity approach for probabilistic bisimulations of diffusion processes. In *Proceedings of the 48th IEEE Conference on Decision and Control and the 28th Chinese Control Conference*, pages 2230–2235, 2009.

[2] A. Abate, J.-P. Katoen, J. Lygeros, and M. Prandini. Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 16(6):1–18, 2010.

[3] A. Abate and M. Prandini. Approximate abstractions of stochastic systems: a randomized method. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, 2011.

[4] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Approximation of general stochastic hybrid systems by switching diffusions with random hybrid jumps. In M. Egerstedt and B. Misra, editors, *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science 4981, pages 598–601. Springer Verlag, 2008.

[5] 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.

[6] D. Adzkiya and A. Abate. Abstraction and verification of autonomous max-plus-linear systems. In *Under review for the 2012 American Control Conference*, 2011.

[7] R. Alur and D.L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.

[8] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity, An Algebra for Discrete Event Systems*. John Wiley and Sons, 1992.

[9] C. Baier, B.R. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time markov chains. *IEEE Trans. Software Eng.*, 29(6):524–541, 2003.

[10] C. Baier, H. Hermanns, J.-P. Katoen, and B.R.H.M. Haverkort. Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *Theor. Comput. Sci.*, 345(1):2–26, 2005.

[11] C. Baier and J.-P. Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[12] R. Bellman. A markovian decision process. *Indiana University Mathematics Journal*, 6:679–684, 1957.

[13] D. P. Bertsekas and S. E. Shreve. *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific, 1996.

[14] P. Billingsley. *Probability and Measure - Third Edition*. Wiley, 1995.

[15] H. Blom. Stochastic hybrid processes with hybrid jumps. In *International Federation of Automatic Control Conference on Analysis and Design of Hybrid Systems*, Saint-Malo, France, 2003.

[16] H.A.P. Blom and J. Lygeros (Eds.). *Stochastic Hybrid Systems: Theory and Safety Critical Applications*. Number 337 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, 2006.

[17] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *J. ACM*, 31(3):560–599, 1984.

[18] M. L. Bujorianu and J. Lygeros. Toward a general theory of stochastic hybrid systems. In H.A.P. Blom and J. Lygeros, editors, *Stochastic Hybrid Systems*, LNCIS 337, pages 3–30. Springer Verlag, 2006.

[19] M.L. Bujorianu and J. Lygeros. General stochastic hybrid systems: Modelling and optimal control. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004.

[20] M.L. Bujorianu, J. Lygeros, and M.C. Bujorianu. Bisimulation for general stochastic hybrid systems. In *Hybrid Systems: Computation and Control (HSCC)*, pages 198–214. Springer-Verlag, 2005.

[21] G. Calafiore and M.C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

[22] M.C. Campi, S. Garatti, and M. Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149–157, 2009.

[23] L. Capra, C. Dutheillet, G. Franceschinis, and J-M. Ili. Exploiting partial symmetries for markov chain aggregation. *Electronic Notes in Theoretical Computer Science*, 39(3):231 – 257, 2000. MTCS 2000 (Satellite Workshop of CONCUR 2000).

[24] C.G. Cassandras and J. Lygeros (Eds.). *Stochastic Hybrid Systems*. Number 24 in Control Engineering. CRC Press, Boca Raton, 2006.

[25] M. H. A. Davis. *Markov Models and Optimization*. Chapman & Hall/CRC Press, London, 1993.

[26] M.H.A. Davis. Piecewise-deterministic markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society*, 46(3):353–384, 1984.

[27] E.P. de Vink and J.J.M.M. Rutten. Bisimulation for probabilistic transition systems: A coalgebraic approach. In *Proceedings of the 24th International Colloquium on Automata Languages and Programming (ICALP 98)*, 1998.

[28] S. Derisavi. Signature-based symbolic algorithm for optimal Markov chain lumping. In *QEST'07*, pages 141–150, 2007.

[29] S. Derisavi. A symbolic algorithm for optimal Markov chain lumping. In *TACAS'07*, pages 139–154, 2007.

[30] S. Derisavi, H. Hermanns, and W.H. Sanders. Optimal state-space lumping in markov chains. *Inf. Process. Lett.*, 87:309–315, September 2003.

[31] S. Derisavi, P. Kemper, and W.H. Sanders. Lumping Matrix Diagram Representations of Markov Models. In *DSN'05*, 2005.

[32] J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labeled Markov processes. *Information and Computation*, 179(2):163–193, Dec. 2002.

[33] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labelled Markov processes. *Theor. Comput. Sci.*, 318(3):323–354, 2004.

[34] J. Desharnais, R. Jagadeesan, V. Gupta, and P. Panangaden. The metric analogue of weak bisimulation for probabilistic processes. In *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS 02)*, pages 413 – 422, 2002.

[35] J. Desharnais, F. Laviolette, and M. Tracol. Approximate analysis of probabilistic processes: logic, simulation and games. In *Proceedings of the International Conference on Quantitative Evaluation of SysTems (QEST '08)*, pages 264–273, Sept. 2008.

[36] J. Desharnais and P. Panangaden. Continuous stochastic logic characterizes bisimulation of continuous-time Markov processes. *J. Log. Algebr. Program.*, 56(1-2):99–115, 2003.

[37] J. Desharnais, P. Panangaden, R. Jagadeesan, and V. Gupta. Approximating labeled markov processes. In *Proceedings of the 15th Annual IEEE Symposium on Logic in Computer Science*, LICS '00, pages 95–105, 2000.

[38] R. Durrett. *Probability: Theory and Examples - Third Edition*. Duxbury Press, 2004.

[39] S.N. Ethier and T.G. Kurtz. *Markov processes: Characterization and convergence*. John Wiley & Sons, 1986.

[40] S. Farahani, T. van den Boom, H. van der Weide, and B. De Schutter. An approximation approach for model predictive control of stochastic max-plus linear systems. In *Proceedings of the 10th International Workshop on Discrete Event Systems (WODES 2010)*, pages 386–391, Berlin, DE, Aug.-Sept. 2010.

[41] P. Florchinger. Lyapunov-like techniques for stochastic stability. *SIAM Journal on Control and Optimization*, 33:1151–1169, July 1995.

[42] M. Fränzle, E.M. Hahn, H. Hermanns, N. Wolovick, and L. Zhang. Measurability and safety verification for stochastic hybrid systems. In *Hybrid Systems: Computation and Control, 13th International Conference, HSCC 2011, Chicago, USA, Proceedings*, pages 43–52, 2011.

[43] M. Fränzle, H. Hermanns, and T. Teige. Stochastic satisfiability modulo theory: A novel technique for the analysis of probabilistic hybrid systems. In Magnus Egerstedt and Bud Mishra, editors, *Proceedings of the 11th International Conference on Hybrid Systems: Computation and Control (HSCC'08)*, volume 4981 of *LNCS*, pages 172–186. Springer, 2008.

[44] M. Fränzle, T. Teige, and A. Eggers. Engineering constraint solvers for automatic analysis of probabilistic hybrid automata. *Journal of Logic and Algebraic Programming*, 79:436–466, 2010.

[45] M. Fränzle, T. Teige, and A. Eggers. Satisfaction meets expectations - computing expected values of probabilistic hybrid systems with SMT. In *IFM*, pages 168–182, 2010.

[46] M. Ghosh and A. Bagchi. Modeling stochastic hybrid systems. In *System Modeling and Optimization. Proceedings of the 21st IFIP TC7 Conference.*, pages 269–279, Sophia Antipolis, FR, 2004.

[47] M. K. Ghosh, A. Araposthasis, and S. I. Marcus. Optimal control of switching diffusions with application to flexible manufacturing systems. *SIAM Journal of Control and Optimization*, 31(5):1183 – 1204, November 1993.

[48] M. K. Ghosh, A. Araposthasis, and S. I. Marcus. Ergodic control of switching diffusions. *SIAM Journal of Control and Optimization*, 35(6):1952–1988, November 1997.

[49] A. Giacalone, C.-C. Jou, and S.A. Smolka. Algebraic reasoning for probabilistic concurrent systems. In *Proceedings of the IFIP TC2 Working Conference on Programming Concepts and Methods*, pages 443–458, 1990.

[50] A. Girard and G.J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Trans. on Automatic Control*, 52(5):782–798, 2007.

[51] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Transactions on Automatic Control*, 55(1):116–126, Jan 2010.

[52] R.M.P. Goverde, B. Heidergott, and G. Merlet. A fast approximation algorithm for the Lyapunov exponent of stochastic max-plus systems. In *Proceedings of the 9th International Workshop on Discrete Event Systems (WODES 2008)*, pages 49–54, may 2008.

[53] T. Han, J.-P. Katoen, and A. Mereacre. Compositional modeling and minimization of time-inhomogeneous Markov chains. In *Hybrid Systems: Computation and Control (HSCC)*, volume 4981 of *LNCS*, pages 244–258. Springer, 2008.

[54] W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, Jul. 2001.

[55] B. Heidergott, G.J. Olsder, and J.W. van der Woude. *Max Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications*. Princeton University Press, 2006.

[56] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. HyTech: A model checker for hybrid systems. *Software Tools for Technology Transfer*, 1:110–122, 1997.

[57] H. Hermanns. *Interactive Markov Chains, and the Quest for Quantified Quality*, volume 2428 of *Lecture Notes in Computer Science*. Springer Verlag, 2002.

[58] H. Hermanns, U. Herzog, and J.-P. Katoen. Process algebra for performance evaluation. *Theor. Comput. Sci.*, 274(1-2):43–87, 2002.

[59] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In H. Hermanns and J. Palsberg, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 3920 of *Lecture Notes in Computer Science*, pages 441–444. Springer Verlag, Berlin Heidelberg, 2006.

[60] D.N. Jansen, H. Hermanns, and J.-P. Katoen. A probabilistic extension of UML Statecharts. In *Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT)*, volume 2469 of *Lecture Notes in Computer Science*, pages 355–374. Springer, 2002.

[61] A.A. Julius and G.J. Pappas. Approximations of stochastic hybrid systems. *Automatic Control, IEEE Transactions on*, 54(6):1193–1203, 2009.

[62] J. Kemeny and J. Snell. *Finite Markov Chains*. D. Van Nostrand, 1960.

[63] J. Kemeny and J. Snell. *Denumerable Markov Chains*. D. Van Nostrand, 1976.

[64] J. Krystul and A. Bagchi. Approximation of first passage time of switching diffusions. In *Proc. MTNS*, Leuwen, BG, July 2004.

[65] J. Krystul, H.A.P. Blom, and A. Bagchi. Stochastic differential equations on hybrid state spaces. In C. Cassandras and J. Lygeros, editors, *Stochastic Hybrid Systems*, volume 24 of *Control Engineering*, pages 15–45. CRC Press, 2006.

[66] H. J. Kushner and P.G. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer-Verlag, New York, 2001.

[67] Marta Z. Kwiatkowska, Gethin Norman, Roberto Segala, and Jeremy Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theor. Comput. Sci.*, 282(1):101–150, 2002.

[68] G. Lafferriere, G.J. Pappas, and S. Sastry. O-minimal hybrid systems. *Mathematics of Control, Signals, and Systems*, 13(1):1–21, 2000.

[69] K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.

[70] W. Lohmiller and J.-J. Slotine. On contraction analysis for nonlinear systems. *Automatica*, 34:671–682, 1998.

[71] R.A. Martin and D. Randall. Disjoint decomposition of Markov chains and sampling circuits in Cayley graphs. *Combinatorics, Probability & Computing*, pages 411–448, 2006.

[72] R. Milner. *A Calculus of Communicating Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.

[73] N. Müllner and O. Theel. Decomposition and reduction of hierarchical self-stabilizing systems. In *submitted to International Conference on Parallel and Distributed Systems (ICPADS)*. OFFIS e.V., IEEE, 2011.

[74] N. Müllner and O. Theel. The Degree of Masking Fault Tolerance vs. Temporal Redundancy. In *Proceedings of the 2011 IEEE 25th International Conference on Advanced Information Networking and Applications Workshops* [75].

[75] N. Müllner and O. Theel. The Degree of Masking Fault Tolerance vs. Temporal Redundancy - Erratum. *http://www.informatik.uni-oldenburg.de/~phoenix/docs/erratum.pdf*, 2011.

[76] B.K. Øksendal. *Stochastic differential equations: an introduction with applications*. Springer Verlag, 2003.

[77] D. Park. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI-Conference on Theoretical Computer Science*, pages 167–183, 1981.

[78] A. Platzer. Stochastic differential dynamic logic for stochastic hybrid programs. In N. Bjørner and V. Sofronie-Stokkermans, editors, *Int'l Conf. on Automated Deduction (CADE)*, 2011.

[79] G. Pola, A. Girard, and P. Tabuada. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10):2508–2516, October 2008.

[80] M. Prandini and J. Hu. Stochastic reachability: Theory and numerical approximation. In C.G. Cassandras and J. Lygeros, editors, *Stochastic hybrid systems*, Automation and Control Engineering Series, pages 107–138. Taylor & Francis Group/CRC Press, 2006.

[81] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley and Sons, New-York, 1994.

[82] S. Sastry. *Nonlinear Systems, Analysis, Stability and Control*. Springer Verlag, 1999.

[83] R. Segala. Probability and nondeterminism in operational models of concurrency. In *17th International Conference on Concurrency Theory (CONCUR)*, volume 4137 of *Lecture Notes in Computer Science*, pages 64–78, 2006.

[84] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995.

[85] R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. *Nord. J. Comput.*, 2(2):250–273, 1995.

[86] J. Sproston. Decidable model checking of probabilistic hybrid automata. In *FTRTFT 2000*, volume 1926 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 2000.

[87] J. Sproston. *Model Checking of Probabilistic Timed and Hybrid Systems*. PhD thesis, University of Birmingham, 2000.

[88] S. Strubbe and A. van der Schaft. Bisimulation for communicating piecewise deterministic Markov processes (CPDPs). In *Hybrid Systems: Computation and Control (HSCC 05)*, pages 623–639. Springer-Verlag, 2005.

[89] T. Teige, A. Eggers, and M. Fränzle. Constraint-based analysis of concurrent probabilistic hybrid systems: An application to networked automation systems. *Nonlinear Analysis: Hybrid Systems*, 5(2):343–366, 2011.

[90] D. Thorsley and E. Klavins. Approximating stochastic biochemical processes with wasserstein pseudometrics. *Systems Biology, IET*, 4(3):193 –211, May 2010.

[91] I. Tkachev and A. Abate. On infinite-horizon probabilistic properties and stochastic bisimulation functions. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, 2011.

[92] F. van Breugel and J. Worrell. Towards quantitative verification of probabilistic transition systems. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming (ICALP 01)*, pages 421–432, 2001.

[93] T. van den Boom and B. De Schutter. Randomly switching max-plus linear systems and equivalent classes of discrete event systems. In *Proceedings of the 9th International Workshop on Discrete Event Systems (WODES 2008)*, pages 242–247, May 2008.

[94] J. van der Woude and B. Heidergott. Asymptotic growth rate of stochastic max-plus systems that with a positive probability have a sunflower-like support. In *Proceedings of the 8th International Workshop on Discrete Event Systems (WODES 2006)*, pages 451–456, 2006.

[95] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics. *J. ACM*, 43(3):555–600, 1996.

[96] A. Xia. Weak convergence of markov processes with extended generators. *The Annals of Probability*, 22:2183–2202, 1994.

[97] L. Zhang, Z. She, S. Ratschan, H. Hermanns, and E.M. Hahn. Safety verification for probabilistic hybrid systems. In *CAV*, volume 6174 of *LNCS*, pages 196–211. Springer Verlag, 2010.

[98] L. Zhang, Z. She, S. Ratschan, H. Hermanns, and E.M. Hahn. Safety verification for probabilistic hybrid systems. *European Journal of Control*, 2011.