Stochastic HYPE: Flow-based modelling of stochastic hybrid systems

Luca Bortolussi
Department of Mathematics and Geosciences
University of Trieste
luca@dmi.units.it

Vashti Galpin Jane Hillston
Laboratory for Foundations of Computer Science
School of Informatics, University of Edinburgh
Vashti.Galpin@ed.ac.uk Jane.Hillston@ed.ac.uk

Stochastic HYPE is a novel process algebra that models stochastic, instantaneous and continuous behaviour. It develops the flow-based approach of the hybrid process algebra HYPE by replacing non-urgent events with events with exponentially-distributed durations and also introduces random resets. The random resets allow for general stochasticity, and in particular allow for the use of event durations drawn from distributions other than the exponential distribution. To account for stochasticity, the semantics of stochastic HYPE target piecewise deterministic Markov processes (PDMPs), via intermediate transition-driven stochastic hybrid automata (TDSHA) in contrast to the hybrid automata used as semantic target for HYPE. Stochastic HYPE models have a specific structure where the controller of a system is separate from the continuous aspect of this system providing separation of concerns and supporting reasoning. A novel equivalence is defined which captures when two models have the same stochastic behaviour (as in stochastic bisimulation), instantaneous behaviour (as in classical bisimulation) and continuous behaviour. These techniques are illustrated via an assembly line example.

1 Introduction

In the last decade there has been increasing interest in capturing stochastic behaviour in models of computational systems. The motivations for this work range from being able to capture probabilistic algorithms, which gain efficiency through non-determinism, to abstraction over an uncertain environment, where a complex variety of possible responses may be represented as a probability distribution rather than a single response. In this paper we introduce stochastic behaviour to HYPE, a formalism previously proposed to model hybrid systems [20]. Our motivation is to capture uncertain response events in a quantified manner.

Hybrid systems mix discrete control and continuous evolution, and can arise in a number of natural and engineering contexts. A number of process algebras for modelling hybrid systems have emerged in recent years including ACP_{hs}^{srt} [3], hybrid χ [2], ϕ-calculus [30] and HyPA [13]. [27] shows in her thesis that there are substantial differences in the approaches taken by these process algebras relating to syntax, semantics, discontinuous behaviour, flow-determinism, theoretical results and availability of tools. Nevertheless there are also strong similarities in their approaches to capturing the continuous evolution of the system: the dynamic behaviour of each continuous variable must be fully described by ordinary differential equations (ODEs) given explicitly in the syntax of the process algebra. Furthermore, they all adopt a notion of state which incorporates an evaluation of continuous variables.

HYPE adopts a different approach which supports a finer-grained view on the dynamics of the system. As it is a process algebra, models are constructed compositionally, but here each subcomponent is itself constructed from a number of *flows* or influences, represented abstractly. Ultimately each flow will correspond to a term in the ODEs governing the evolution of a continuous variable, but at the process

algebra level the details of the mathematical form of this influence are left unspecified. This separation of concerns allows us to consider the logical structure of the model (the process algebra description) as distinct from any particular dynamic realisation. We believe that the use of flows as the basic elements of model construction has advantages such as ease and simplification of modelling. This approach assists the modeller, allowing them to identify smaller or local descriptions of the model and then to combine these descriptions to obtain the larger system.

Discrete changes within a system are captured as *events*. Each event may change the dynamic nature of a flow, possibly also resetting the current value of continuous variables. Each event has a triggering condition which may be dependent on the current value of the continuous variables. In HYPE as presented by [20], events are either *urgent* or *non-urgent*. Urgent events are forced to occur as soon as their trigger becomes true, and if multiple urgent events become true simultaneously, one of them occurs, and the remaining events may or may not occur after this event depending on whether their trigger condition remains true. In contrast, non-urgent actions may delay indefinitely.

HYPE can be viewed as belonging to the class of quantified process algebras, including probabilistic, stochastic and timed process algebras which support reasoning over models to quantitatively analyse the behaviour of a system, for example considering the relative likelihood of particular sub-behaviours, or the expected time until a condition is met. However, the inclusion of non-urgent events within the model left some unspecified behaviour that could not be quantified and therefore limited the extent to which quantified analysis could be applied. Thus in this paper we define an extension of HYPE, stochastic HYPE, in which the non-urgent events of HYPE are replaced by stochastic events. Such events remain non-urgent, as in not occurring immediately, but now their delay is governed by an exponential distribution. Additionally through the use of random variables in resets, other forms of stochasticity can be included in models, and timers can be used to achieve random delays from an arbitrary distribution. So the non-urgent events of HYPE are effectively replaced in stochastic HYPE by random events based on any distribution. The underlying mathematical model used by stochastic HYPE is Transition Driven Stochastic Hybrid Automata (TDSHA) [6], based on piecewise deterministic Markov process (PDMP) [15]. We work with TDSHA because they are a better than operational match with stochastic HYPE than PDMPs.

The contributions of this paper include a compositional language to describe the dynamics of systems with stochastic, instantaneous and continuous behaviour, in a modular fashion to ensure models are straightforward to modify. This requires mapping to a mathematical model of these three behaviour types to TDSHA, related to PDMP. Furthermore, a new bisimulation is defined to enable formal reasoning about differences and similarities in model behaviour. These contributions are illustrated through a non-trivial example of an assembly line.

The remainder of this paper is structured as follows. In Section 2 we illustrate the stochastic HYPE modelling language, through the description of a simple network example, and the formal syntax of stochastic HYPE. The operational semantics is defined in terms of *configurations* which capture the influences at play within the system rather than explicit values of continuous variables, is presented in Section 3. The dynamic interpretation of stochastic HYPE models is given in terms of a class of hybrid automata called Transition-Driven Stochastic Hybrid Automata (TDSHA) [6] which can themselves be mapped to PDMPs [15]. The TDSHA formalism is presented in Section 4 and the mapping from stochastic HYPE to TDSHA is defined in Section 5. This section also considers the impact of stochastic events on the notion of well-behavedness. A HYPE model is well-behaved if it cannot execute an infinite number of simultaneous instantaneous events. Introducing stochastic events to a model that was previously not well-behaved, can lead to well-behavedness. Section 6 introduces the main example of the paper and illustrates the expressiveness of stochastic HYPE through an optimisation problem. Semantic

$$
\begin{array}{rcl}
\mathit{Input} & \stackrel{def}{=} & \overline{\mathrm{on}}_{in}:(in,r_{in},const).\mathit{Input}+\overline{\mathrm{off}}_{in}:(in,0,const).\mathit{Input}+ \\
 & & \underline{\mathrm{full}}:(in,0,const).\mathit{Input}+\underline{\mathrm{init}}:(in,0,const).\mathit{Input} \\
\mathit{Output} & \stackrel{def}{=} & \overline{\mathrm{on}}_{out}:(out,-r_{out},const).\mathit{Output}+\overline{\mathrm{off}}_{out}:(out,0,const).\mathit{Output}+ \\
 & & \underline{\mathrm{empty}}:(out,0,const).\mathit{Output}+\underline{\mathrm{init}}:(out,0,const).\mathit{Output} \\
\mathit{Drop} & \stackrel{def}{=} & \underline{\mathrm{init}}:(f,0,const).\mathit{Drop}+\underline{\mathrm{fail}}:(f,0,const).\mathit{Drop} \\
\mathit{Timer} & \stackrel{def}{=} & \underline{\mathrm{init}}:(t,1,const).\mathit{Timer}
\end{array}
$$

$$
\begin{array}{rcl}
\mathit{Con}_{in} & \stackrel{def}{=} & \overline{\mathrm{on}}_{in}.\mathit{Con}'_{in} \qquad \mathit{Con}'_{in} \stackrel{def}{=} \overline{\mathrm{off}}_{in}.\mathit{Con}_{in}+\underline{\mathrm{full}}.\mathit{Con}_{in} \\
\mathit{Con}_{out} & \stackrel{def}{=} & \overline{\mathrm{on}}_{out}.\mathit{Con}'_{out} \quad \mathit{Con}'_{out} \stackrel{def}{=} \overline{\mathrm{off}}_{out}.\mathit{Con}_{out}+\underline{\mathrm{empty}}.\mathit{Con}_{out} \\
\mathit{Con}_{fail} & \stackrel{def}{=} & \underline{\mathrm{fail}}.\mathit{Con}_{fail}
\end{array}
$$

$$
\begin{array}{rcl}
\mathit{Buffer} & \stackrel{def}{=} & (\mathit{Input} \bowtie_{*} \mathit{Output} \bowtie_{*} \mathit{Drop} \bowtie_{*} \mathit{Timer}) \bowtie_{*} \underline{\mathrm{init}}.(\mathit{Con}_{in} \parallel \mathit{Con}_{out} \parallel \mathit{Con}_{fail}) \\
\mathscr{V} & = & \{B,T,C,D\} \quad iv(in)=B \quad iv(out)=B \quad iv(f)=B \quad iv(t)=T
\end{array}
$$

$$
\begin{array}{rcl}
ec(\underline{\mathrm{init}}) & = & (true,\ B'\sim b_0 \wedge T'\sim 0 \wedge C'\sim 0 \wedge D'\sim \ln \mathscr{N}(\Delta,\xi)) \\
ec(\underline{\mathrm{fail}}) & = & (T=C+D,\ C'\sim T \wedge D'\sim \ln \mathscr{N}(\Delta,\xi)) \wedge B'\sim B-\mathscr{U}(0,B)) \\
ec(\underline{\mathrm{full}}) & = & (B=max_B,true) \qquad ec(\underline{\mathrm{empty}}) = (B=0,true) \\
ec(\overline{\mathrm{on}}_{in}) & = & (k^{on}_{in},\ true) \qquad ec(\overline{\mathrm{off}}_{in}) = (k^{off}_{in},\ true) \\
ec(\overline{\mathrm{on}}_{out}) & = & (k^{on}_{out},\ true) \qquad ec(\overline{\mathrm{off}}_{out}) = (k^{off}_{out},\ true)
\end{array}
$$

Figure 1: Simple network node model in stochastic HYPE.

equivalences play an important role in reasoning about process algebra models, and a major contribution of this paper is a suitable bisimulation for stochastic HYPE models. In Section 7, we show why existing bisimulations are not sufficient and present a bisimulation for stochastic HYPE that captures the notions that two models share the same behaviours for all three behaviour types. In Section 8, the use of this bisimulation and associated results are demonstrated on the example from Section 6. Then Section 9 presents related work and we conclude in Section 10. Online appendices contain proofs and supplementary material.

Preliminary work on stochastic HYPE has been published by [4]; here we define a more elegant mapping to TDSHA, add random resets and present new results on bisimulation equivalences.

2 Stochastic HYPE Definition

In this section we present the definition of stochastic HYPE by means of a small example. More details about the process algebra HYPE can be found in an earlier paper [20]. We consider a basic model of a network node with a single buffer, which can either receive packets from an input channel or send packets to an output channel. We assume that the number of packets that travel through the node and that are stored in the buffer is large, hence we describe them as a fluid quantity. Received packets are stored in the buffer, waiting to be sent. We allow reception and sending of packets to happen concurrently, but enforcing a mutually exclusive send/receive policy can be easily done as well. We also assume that

uplinks and downlinks are not always working, but they are activated and deactivated depending on the availability of a connection. These events are described as stochastic, with activation and deactivation times governed by exponential distributions. Incoming traffic has to be stopped if the buffer becomes full and outgoing traffic has to be stopped when the buffer is empty. The buffer also shows intermittent error behaviour in that after the passage of a random time period, (drawn from a log normal distribution with fixed mean and variance), it drops some of its packets (this quantity is determined uniformly over the number of packets).

HYPE modelling is centred around the notion of *flow*, which represents some sort of influence continuously modifying one quantity of the system, described by a variable taking continuous values. Both the strength and form of a flow can be changed by the happening of *events*. In our example, there are two flows modifying the buffer level, modelled by the continuous variable B, namely reception and sending of packets. The inflow of packets is modelled by the *Input subcomponent* shown in Figure 1. Each subcomponent is a summation of prefixes consisting of an *event* a followed by an *activity* α. *Events* ($a \in \mathscr{E}_d \cup \mathscr{E}_s$) are actions which trigger discrete changes. They can be caused by a controller, which triggers them depending on the global state of the system, specifically on values of variables (instantaneous events $\underline{a} \in \mathscr{E}_d$), or happen at exponentially distributed time instants (stochastic events $\overline{a} \in \mathscr{E}_s$). In the description of the example, non-exponential durations are mentioned but these are constructed explicitly with timers. We show later in the paper how we use a syntactic shorthand for events that have a duration from any distribution.

The *Input* subcomponent reacts to three different events: two stochastic ones, activation $\overline{\text{on}}_{in}$ and deactivation $\overline{\text{off}}_{in}$ of the uplink, and one instantaneous event, $\underline{\text{full}}$, triggered when the buffer level is at its maximum capacity. The event $\underline{\text{init}}$ serves to initialize the system.

Activities ($\alpha \in \mathscr{A}$) are *influences* on the evolution of the continuous part of the system and define flows. An activity is defined as a triple and can be parameterised by a set of variables, $\alpha(\mathscr{W}) = (\iota, r, I(\mathscr{W}))$. This triple consists of an *influence name* ι, a rate of change (or *influence strength*) r and an *influence type name* $I(\mathscr{W})$ which describes how that rate is to be applied to the variables involved, or the actual form of the flow[1]. In *Input*, there are two distinct activities: $(in, r_{in}, const)$ and $(in, 0, const)$. The first one gives the effect of the incoming link being active on the buffer level B: the influence name is *in*, which uniquely identifies the effect of the input link on B, r_{in} is the strength of the inflow of packets (here it can be seen as the amount of received data per time unit), which is associated with the function *const*. The second activity captures the effect of the link being down. It again affects influence *in*, but has strength 0 and the form it takes is again *const*.

The interpretation of influence types will be specified separately, in order to experiment with different functional forms of the packet inflow without modifying the subcomponent. In this case, we will obviously interpret *const* as the constant function 1. Hence, in HYPE we separate the description of the logical structure of flows from their mathematical interpretation.

A second subcomponent affecting buffer level is the output component *Output* in Figure 1, modelling the sending of packets, which is defined similarly to *Input*. The third subcomponent *Drop* has no effect on the contents of the buffer, since its influence strength is always 0, but it does introduce the event $\underline{\text{fail}}$ that corresponds to the dropping of packets. This event will be described below. Finally, since the dropping of packets occurs with a frequency, a time subcomponent and a time variable are required to describe the passing of time. It is not affected by any event apart from $\underline{\text{init}}$. These subcomponents can be combined to give the overall uncontrolled system

$$\mathit{Input} \underset{*}{\bowtie} \mathit{Output} \underset{*}{\bowtie} \mathit{Drop} \underset{*}{\bowtie} \mathit{Timer}$$

[1] For convenience, we will use I for $I(\mathscr{W})$ when $\mathscr{W}$ can be inferred.

Here $\bowtie_*$ represents parallel cooperation where all shared events must be synchronised (we use $\bowtie_L$ to specify that the events in L are to be synchronised on and $\|$ when no events are to be synchronised on). This cooperation of subcomponents is called the *uncontrolled system* because it only specifies how flows react to events, without imposing any causal or temporal constraints on them.

Causality on events, reflecting natural constraints or design choices, is specified separately in the controller *Con*. For the system at hand, the controller is defined as shown in Figure 1. The subcontroller Con_{in} models the fact that the reception of packets can be turned off only after being turned on. Furthermore, it describes termination of the input if the buffer becomes full, but only if the uplink is active. Con_{out} is similar. The $\underline{\text{fail}}$ event happens without reference to other events, and is determined by a timer as will be shown below. However, a simple controller is added for this event so that the same events appear both in the controller and the uncontrolled system.

The uncontrolled system and the controller are then combined together to define the *controlled system Buffer*.

Controllers have only event prefixes, and we need to specify when events are activated. This is achieved by assigning to each event a set of event conditions, which differ between *stochastic* and *deterministic* events.

Deterministic events $\underline{\text{a}} \in \mathscr{E}_d$ happen when certain conditions are met by the system. These *event conditions* are specified by a function *ec*, assigning to each event a *guard* or *activation condition* (a boolean combination of predicates depending on system variables, stating when a transition can fire) and a *reset* (specifying how variables are modified by the event). For example, $ec(\underline{\text{full}}) = (B = max_B, true)$ states that the uplink is shut down when the buffer reaches its maximum capacity max_B, and no variable is modified. An event condition that involves resets is that of the event $\underline{\text{fail}}$[2].

$$ec(\underline{\text{fail}}) \quad = \quad (T = C+D,\, C' \sim T \wedge D' \sim \ln \mathscr{N}(\Delta, \xi)) \wedge B' \sim B - \mathscr{U}(0,B))$$

The activation condition requires the time variable to be the sum of C and D, and when this is true, C is assigned the value of the current time, and a new random duration D is drawn from a log normal distribution with mean Δ time units and variance ξ thus determining how long until the next failure. The value of B is also decreased by a random variable drawn from the uniform distribution for all non-negative values less than or equal to B and this modification of variable B represents the dropping of packets.

The symbol $\sim$ indicates that there may be random values in the reset. When no distribution is indicated, such as $C' \sim T$, this is equivalent to $C' = T$. Multiple resets are combined using $\wedge$ indicating that all resets must occur. The notation V' is used to denote the value of V after the reset occurred. Deterministic events in HYPE are *urgent*, meaning that they fire as soon as their guard becomes true.

Stochastic events $\overline{\text{a}} \in \mathscr{E}_s$ have an event condition composed of a stochastic rate (replacing the guard of deterministic events) and a reset the same as above. For instance, $ec(\overline{\text{on}}_{in}) = (k_{in}^{on}, true)$ states that the duration of packet reception is a stochastic event happening at times exponentially distributed with constant rate k_{in}^{on}. Rates define exponential distributions and can be functions of the variables of the system. We choose to restrict delays for stochastic events to those that are exponential distributed to avoid the issue of residual clocks when there is no explicit timer associated with the delays. The event condition for $\underline{\text{fail}}$ shows how delays with other distributions can be modelled with the use of timers, and we generalise this approach later.

[2]In the original definition of HYPE, resets were deterministic and were defined using standard functional notation. This notation must be modified to allow for random resets and is defined formally later in this section.

We also need to link each influence with an actual variable. This is done using the function iv. For the example, $iv(in) = iv(out) = iv(f) = B$, where B is the buffer level. The same variable can be modified by many influences. Note also that in order to model situations in which an event modifies the flow of more than one variable (in the example, the activation of a link might modify, for instance, battery consumption), we can simply define more subcomponents and combine them into structured components, synchronizing them on shared events.

Finally, we need to interpret influence types, mapping them to proper functions. In the example, we set $[\![const]\!] = 1$, so that *const* defines a constant flow. However, influence types can be mapped to linear and non-linear functions as well, as shown in other HYPE examples [21, 20].

In the preceding informal discussion, we have introduced the main constituents of a stochastic HYPE model including the combination of flow components with a controller component, variables, association between influences and variables, conditions that specify when events occur, and definitions for the influence type functions. To understand the dynamics of this system, we need to derive ODEs to describe how the variables change over time, and to further specify how the discrete and the continuous dynamics interact. We will do this by defining an operational semantics, which will specify qualitatively the behaviour of a controlled system, and then mapping the so-obtained labelled transition system into a special class of Stochastic Hybrid Automata. Before that we give the formal definition of stochastic HYPE. In the rest of the paper, $\mathscr{V}$ is a set or tuple of variables with $\mathscr{W} \subseteq \mathscr{V}$ denoting an arbitrary subset of $\mathscr{V}$.

Definition 1. *A* stochastic HYPE model *is a tuple* $(ConSys, \mathscr{V}, IN, IT, \mathscr{E}_d, \mathscr{E}_s, \mathscr{A}, ec, iv, EC, ID)$ *where*

- *ConSys is a well-defined controlled system as defined below.*
- $\mathscr{V}$ *is a finite set of variables.*
- *IN is a set of influence names and IT is a set of influence type names.*
- $\mathscr{E}_d$ *is the set of instantaneous events of the form* $\underline{\mathrm{a}}$ *and* $\underline{\mathrm{a}}_i$*.*
- $\mathscr{E}_s$ *is the set of stochastic events of the form* $\overline{\mathrm{a}}$ *and* $\overline{\mathrm{a}}_i$*.*
- $\mathscr{E} = \mathscr{E}_d \cup \mathscr{E}_s$ *is the set of all events for which the notation* a *and* a_i *is used.*
- $\mathscr{A}$ *is a set of activities of the form* $\alpha(\mathscr{W}) = (\iota, r, I(\mathscr{W})) \in (IN \times \mathbb{R} \times IT)$ *where* $\mathscr{W} \subseteq \mathscr{V}$*.*
- $ec : \mathscr{E} \to EC$ *maps events to event conditions. Event conditions are pairs of activation conditions and resets.*
  - *Activation conditions for the instantaneous events in* $\mathscr{E}_d$ *are formulae with free variables in* $\mathscr{V}$ *and for the stochastic events in* $\mathscr{E}_s$*, they are functions* $f : \mathbb{R}^{|\mathscr{V}|} \to \mathbb{R}^+$*.*
  - *Resets are conjunctions of formulae of the form* $V' \sim \theta(\mathscr{V})$ *where* $\theta(\mathscr{V})$ *is an expression that may also include random variables which are described by* $\mathscr{X}(p_1, \ldots, p_m)$ *where* $\mathscr{X}$ *is a distribution and the* p_i *are the parameters for the distribution which can themselves be expressions involving variables and random variables.*
- $iv : IN \to \mathscr{V}$ *maps influence names to variable names.*
- *EC is a set of event conditions.*
- *ID is a collection of definitions consisting of a real-valued function for each influence type name* $[\![I(\mathscr{W})]\!] = f(\mathscr{W})$ *where the variables in* $\mathscr{W}$ *are from* $\mathscr{V}$*.*
- $\mathscr{E}$*,* $\mathscr{A}$*, IN and IT are pairwise disjoint.*

In [20], *well-defined HYPE models* are introduced and the rationale behind the definition is to enforce a policy in the way models are specified, forcing the modeller to separate the description of the logical blocks constituting a HYPE model. Subcomponents are "flat" (i.e. they always call themselves recursively) and there is a one-to-one correspondence between influences and subcomponents: each subcomponent describes how a specific influence is modified by events. Furthermore, synchronization must involve all shared events. This notion extends straightforwardly to stochastic HYPE models, and here we choose to define stochastic HYPE systems in their well-defined form immediately rather than having separate definitions.

Definition 2. *A* well-defined controlled system *has the form* $\Sigma \bowtie_{*} \underline{\text{init}}.Con$ *where* Σ *is a well-defined uncontrolled system and Con is a well-defined controller, and the synchronisation is on all shared events, in particular* $\underline{\text{init}}$*, the initial event which must occur first (and has true as its activation condition). Furthermore, all events that appear in the controller must appear in the uncontrolled system and* vice versa*. The set of well-defined controlled systems is denoted by* $\mathscr{C}$*.*

A well-defined controller *is defined by the two-level grammar* $M ::= \mathrm{a}.M \mid 0 \mid M+M$ *and* $Con ::= M \mid Con \bowtie_{L} Con$ *with* $\mathrm{a} \in \mathscr{E}$ *and with* $L \subseteq \mathscr{E}$*.*

The well-defined uncontrolled system *consists of well-defined subcomponents in cooperation over shared events where each subcomponent appears at most once and* $\mathscr{W}_i \subseteq \mathscr{V}$*.*

$$\Sigma \stackrel{\text{def}}{=} S_1(\mathscr{W}_1) \bowtie_{*} \ldots \bowtie_{*} S_s(\mathscr{W}_s)$$

Well-defined subcomponents *represent the uncontrolled capabilities of the system and each is a choice over events such that* $\mathrm{a}_j \neq \mathrm{a}_k$ *for* $j \neq k$*,* $\mathrm{a}_j \neq \underline{\text{init}}$ *for all j, and* $\mathscr{W} \subseteq \mathscr{V}$*.*

$$S(\mathscr{W}) \stackrel{\text{def}}{=} \sum_{j=1}^{n} \mathrm{a}_j{:}(\iota, r_j, I_j(\mathscr{W})).S(\mathscr{W}) + \underline{\text{init}}{:}(\iota, r, I(\mathscr{W})).S(\mathscr{W})$$

A subcomponent is ready to react whenever any of its events' activation condition becomes true or completes, after which the influence associated with that event comes into force, replacing any previous influence. By considering all the influences mapped to a particular variable for a particular configuration of the system, an ODE can be constructed using the definitions in *ID* to describe the evolution of that variable whenever the system is in that configuration. We assume well-defined stochastic HYPE models in the rest of this paper. With the syntax of stochastic HYPE models defined, the next three sections show how the dynamic behaviour of stochastic HYPE models can be defined before illustrating this dynamic behaviour with an substantial example of an assembly line.

3 Operational Semantics

We now introduce the behaviour of stochastic HYPE models by defining an appropriate semantics. We will proceed in a different way to that presented in the preliminary research [4]: instead of mapping each subcomponent and each controller to a stochastic hybrid automaton, and then composing them together with a product construction, we will construct a labelled (multi)transition system (LTS), similarly to [24], and then map the LTS to a stochastic hybrid automaton.

To construct the LTS, we need a notion of state. Proceeding in the same manner as HYPE [20], states will record for each influence its current strength and influence type. States essentially capture the continuous behaviour, while the structure of the LTS describes the discrete behaviour, both instantaneous and stochastic.

Prefix with influence:

$$\frac{}{\left\langle \mathrm{a} : (\iota, r, I).P, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle P, \sigma[\iota \mapsto (r, I)] \right\rangle} (\mathrm{a} \in \mathscr{E})$$

Prefix without influence:

$$\frac{}{\left\langle \mathrm{a}.P, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle P, \sigma \right\rangle} (\mathrm{a} \in \mathscr{E})$$

Choice:

$$\frac{\left\langle P, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle P', \sigma' \right\rangle}{\left\langle P + Q, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle P', \sigma' \right\rangle} \qquad \frac{\left\langle Q, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle Q', \sigma' \right\rangle}{\left\langle P + Q, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle Q', \sigma' \right\rangle}$$

Constant:

$$\frac{\left\langle P, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle P', \sigma' \right\rangle}{\left\langle A, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle P', \sigma' \right\rangle} (A \stackrel{def}{=} P)$$

Cooperation without synchronisation:

$$\frac{\left\langle P, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle P', \sigma' \right\rangle}{\left\langle P \underset{L}{\bowtie} Q, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle P' \underset{L}{\bowtie} Q, \sigma' \right\rangle} (\mathrm{a} \notin L) \qquad \frac{\left\langle Q, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle Q', \sigma' \right\rangle}{\left\langle P \underset{L}{\bowtie} Q, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle P \underset{L}{\bowtie} Q', \sigma' \right\rangle} (\mathrm{a} \notin L)$$

Cooperation with synchronisation:

$$\frac{\left\langle P, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle P', \tau \right\rangle \quad \left\langle Q, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle Q', \tau' \right\rangle}{\left\langle P \underset{L}{\bowtie} Q, \sigma \right\rangle \xrightarrow{\mathrm{a}} \left\langle P' \underset{L}{\bowtie} Q', \Gamma(\sigma, \tau, \tau') \right\rangle} (\mathrm{a} \in L, \Gamma \text{ defined})$$

Figure 2: Operational semantics for stochastic HYPE

Definition 3. *An* operational state *is a function* $\sigma : IN \to (\mathbb{R} \times IT)$. *The set of all operational states is* $\mathscr{S}$. *A* configuration *consists of a controlled system together with an operational state* $\left\langle ConSys, \sigma \right\rangle$ *and the set of configurations is* $\mathscr{F}$.

In the following, we will usually refer to 'operational states' as 'states'. States can also be viewed as a set of triples $(\iota, r, I(\mathscr{W}))$, where there is at most one triple containing ι. In this form, it is evident that states are simply collections of enabled activities. We stress that states describe flows, not the values of continuous variables. As such, the LTS semantics of stochastic HYPE is different to that of stochastic hybrid automata, similarly to the way that the LTS semantics of HYPE differs from those of hybrid automata [20].

The operational semantics give a labelled multitransition system over configurations $(\mathscr{F}, \mathscr{E}, \to \subseteq \mathscr{F} \times \mathscr{E} \times \mathscr{F})$. As customary, we write $E \xrightarrow{\mathrm{a}} F$, for $E, F \in \mathscr{F}$. The rules are given in Figure 2 and are essentially those of non-stochastic HYPE. The only formal difference is that we consider in the rules both instantaneous and stochastic events, treating them in the same way. The fact that the operational semantic rules are essentially the same is because the LTS describes the syntactic structure of the target hybrid automaton. The proper dynamics will be attached to a (stochastic) HYPE model by converting its LTS to a (stochastic) hybrid automaton. This second step of the semantics construction differs significantly from that used for HYPE [20].

Inspecting the rules of Figure 2, we can see that the only rules updating the state are Prefix with influence and Cooperation with synchronisation.

In this latter case, in particular, we need to enforce consistency in the way influences are updated by the cooperating components. This is done by the function Γ. The updating function $\sigma[\iota \mapsto (r, I)]$ is defined as

$$\sigma[\iota \mapsto (r, I)](x) = \begin{cases} (r, I) & \text{if } x = \iota \\ \sigma(x) & \text{otherwise.} \end{cases}$$

The notation $\sigma[u]$ will also be used for an update, with $\sigma[u_1 \dots u_n]$ denoting $(\dots((\sigma[u_1])[u_2])\dots)[u_n]$.

The partial function $\Gamma : \mathscr{S} \times \mathscr{S} \times \mathscr{S} \to \mathscr{S}$ is defined as follows.

$$(\Gamma(\sigma,\tau,\tau'))(\iota) \quad = \quad \begin{cases} \tau(\iota) & \text{if } \sigma(\iota) = \tau'(\iota), \\ \tau'(\iota) & \text{if } \sigma(\iota) = \tau(\iota), \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Γ is undefined if both agents in a cooperation try to update the same influence. See [20] for further details. In the following, we will also need some additional notions.

Definition 4. *Given a controlled system P, define structurally the following sets:*

- *the* set of events, $\mathsf{ev}(P)$*:* $\mathsf{ev}(\mathrm{a}:\alpha.S) = \{\mathrm{a}\}$, $\mathsf{ev}(\mathrm{a}.S) = \{\mathrm{a}\}$, $\mathsf{ev}(S_1 + S_2) = \mathsf{ev}(S_1) \cup \mathsf{ev}(S_2)$, $\mathsf{ev}(P_1 \underset{L}{\bowtie} P_2) = \mathsf{ev}(P_1) \cup \mathsf{ev}(P_2)$*;*
- *the* set of influences, $\mathsf{in}(P)$*:* $\mathsf{in}(\mathrm{a}:(\iota,r,I(\mathscr{W})).S) = \{\iota\}$ $\mathsf{in}(\mathrm{a}.S) = \emptyset$, $\mathsf{in}(S_1 + S_2) = \mathsf{in}(S_1) \cup \mathsf{in}(S_2)$, $\mathsf{in}(P_1 \underset{L}{\bowtie} P_2) = \mathsf{in}(P_1) \cup \mathsf{in}(P_2)$*;*
- *the* set of prefixes*:* $\mathsf{pr}(\mathrm{a}:\alpha.S) = \{\mathrm{a}:\alpha\}$, $\mathsf{pr}(\mathrm{a}.S) = \emptyset$, $\mathsf{pr}(S_1 + S_2) = \mathsf{pr}(S_1) \cup \mathsf{pr}(S_2)$, $\mathsf{pr}(P_1 \underset{L}{\bowtie} P_2) = \mathsf{pr}(P_1) \cup \mathsf{pr}(P_2)$.

Definition 5. *The* derivative set *of a controlled system P,* $\mathrm{ds}(P)$*, is defined as the smallest set satisfying*

- *if* $\langle P,\sigma\rangle \xrightarrow{\text{init}} \langle P',\sigma'\rangle$ *then* $\langle P',\sigma'\rangle \in \mathrm{ds}(P)$
- *if* $\langle P',\sigma'\rangle \in \mathrm{ds}(P)$ *and* $\langle P',\sigma'\rangle \xrightarrow{\mathrm{a}} \langle P'',\sigma''\rangle$ *then* $\langle P'',\sigma''\rangle \in \mathrm{ds}(P)$.

Furthermore, we indicate with $\mathrm{st}(P) = \{\sigma \mid \langle Q,\sigma\rangle \in \mathrm{ds}(P)\}$ *the set of* states *of the derivative set of a controlled system P.*

In the *Buffer* model, there are four states in the LTS, corresponding to the four possible combinations of activation of input and output channels. The influences f and t are the same in all states. For instance, the state in which the input link is active and the output is not active is

$$\sigma \quad = \quad \{in \mapsto (r_{in}, const),\ out \mapsto (0, const),\ f \mapsto (0, const),\ t \mapsto (1, const)\}$$

Propositions 5.1 to 5.5 proved for HYPE [20], hold for well-defined stochastic HYPE models. In particular, the function Γ used in the Cooperation with synchronisation rule, is always defined for well-defined stochastic HYPE systems. The *Buffer* system is well-defined, and hence Γ is always defined.

4 Transition-Driven Stochastic Hybrid Automata

Because stochastic HYPE considers three distinct types of behaviour, whereas HYPE considers only two, we need a different target for our semantics. The semantics for HYPE are provided by hybrid automata which covers continuous and instantaneous behaviour but not stochastic behaviour. We now present Transition-Driven Stochastic Hybrid Automata, introduced by Bortolussi and Policriti [6, 7], a formalization of stochastic hybrid automata putting emphasis on transitions, which can be either discrete (corresponding to instantaneous or stochastic jumps) or continuous (representing flows acting on system variables). This formalism can be seen as an intermediate layer in defining the stochastic hybrid semantics of stochastic HYPE. In fact, TDSHA are themselves mapped to Piecewise Deterministic Markov Processes [15], so that their dynamics can be formally specified in terms of the latter. Due to space constraints, we will not provide a formal treatment of this construction, and refer the reader papers by Bortolussi and Policriti [6, 7] and [8] for further details. We choose to work with TDSHA rather than PDMPs because their definition is more operational in nature, and hence this leads to a more straightforward mapping.

Definition 6. *A Transition-Driven Stochastic Hybrid Automaton (TDSHA) is a tuple* $\mathscr{T} = (Q, \mathbf{X}, \mathscr{TC}, \mathscr{TD}, \mathscr{TS}, \text{init}, \mathscr{E})$*, where*

- Q *is a finite set of* control modes*.*
- $\mathbf{X} = \{X_1, \ldots, X_n\}$ *is a set of real valued* system variables *with the time derivative of* X_j *denoted by* $\dot{X}_j$ *and the value of* X_j *after a change of mode denoted by* X'_j*.*
- $\mathscr{TC}$ *is the* multiset *of* continuous transitions or flows*, whose elements* τ *are triples* $(q_\tau, \mathbf{s}_\tau, f_\tau)$*, where* $q_\tau \in Q$ *is a mode,* $\mathbf{s}_\tau$ *is a vector of size* $|\mathbf{X}|$*, and* $f_\tau : \mathbb{R}^n \to \mathbb{R}$ *is a* Lipschitz continuous *function.*
- $\mathscr{TD}$ *is the* set *of* instantaneous transitions*, whose elements* δ *are tuples of the form* $(q_1^\delta, q_2^\delta, g_\delta, r_\delta, w_\delta, e_\delta)$*. The transition goes from mode* q_1^δ *to mode* q_2^δ*.*
  - *The* guard g_δ *is a first-order formula with free variables from* $\mathbf{X}$*, representing the* closed set $G_\delta = \{\mathbf{x} \in \mathbb{R}^n \mid g[\mathbf{x}]\}$*.*
  - *The* reset r_δ *is a conjunction of atoms of the form* $X' = r(\mathbf{X}, \mathbf{W})$ *where* $r : \mathbb{R}^n \times \mathbb{R}^h \to \mathbb{R}$ *is the reset function of* X *dependent on the variables* $\mathbf{X}$ *as well as a random vector* $\mathbf{W}$*. Variables not appearing in the reset remain unchanged and a reset with value true is the identity function on each variable.*
  - *The* weight (priority) *of the edge is* $w_\delta \in \mathbb{R}^+$ *and is used to solve non-determinism among two or more active transitions.*
  - *The* label *of the edge is* $e_\delta \in \mathscr{E}$*.*
- $\mathscr{TS}$ *is the* multiset *of* stochastic transitions*, whose elements* η *are tuples of the form* $\eta = (q_1^\eta, q_2^\eta, g_\eta, r_\eta, f_\eta, e_\eta)$*, where* q_1^η*,* q_2^η*,* g_η*,* e_η*, and* r_η *are as for transitions in* $\mathscr{TD}$*.*
  - *The* rate *of the edge is* $f_\eta : \mathbb{R}^n \to \mathbb{R}^+$*, a rate function giving the instantaneous probability of taking transition* η*. It is locally* integrable *along any continuous differentiable curve. Additionally, transitions labelled by the same event are required to have* consistent *rates: if* $e_{\eta_1} = e_{\eta_2}$*, then* $f_{\eta_1} = f_{\eta_2}$*.*
- $\mathscr{E}$ *is a finite set of event names, labelling discrete transitions.* $\mathscr{E}$ *can be partitioned into* $\mathscr{E}_d \cup \mathscr{E}_s$*, such that all events labelling instantaneous transitions belong to* $\mathscr{E}_d$*, while all events labelling stochastic transitions are from* $\mathscr{E}_s$*.*
- init *is a pair* $(q^{init}, \mathbf{W})$*, with* $q^{init} \in Q$ *and* $\mathbf{W}$ *is a random vector of n variables representing a point in* $\mathbb{R}^n$ *and providing the initial values for* $\mathbf{X}$*.* init *describes the initial state of the system.*

Note that in the previous definition, we consider sets of instantaneous transitions and multisets of stochastic and continuous transitions. This is justified by the fact that multiplicity plays a relevant role only in quantified behaviours (i.e. flows and stochastic events), but it is not really relevant for instantaneous events, which are triggered by a boolean condition, whose truth is insensitive to the presence of multiple transitions of the same kind.

In order to formally define the dynamical evolution of TDSHA, we can map them into a well-studied model of Stochastic Hybrid Automata, namely Piecewise Deterministic Markov Processes [15]. Here we sketch basic ideas about the dynamical behaviour of TDSHA.

- Within each discrete mode $q \in Q$, the system follows the solution of a set of ODEs, constructed combining the effects of the continuous transitions τ acting on mode q. The function $f_\tau(\mathbf{X})$ is multiplied by the vector $\mathbf{s}_\tau$ to determine its effect on each variable and then all such functions are added together, so that the ODEs in mode q are $\dot{\mathbf{X}} = \sum_{\tau \mid q_\tau = q} \mathbf{s}_\tau \cdot f_\tau(\mathbf{X})$.

- Two kinds of discrete jumps are possible. Stochastic transitions are fired according to their rate, similarly to standard Markovian Jump Processes. Instantaneous transitions, by contrast, are fired as soon as their guard becomes true. In both cases, the state of the system is reset according to the specified reset policy which is not necessarily deterministic. Choice among several active stochastic or instantaneous transitions is performed probabilistically proportionally to their rate or weight.
- A trace of the system is therefore a sequence of instantaneous and random jumps interleaved by periods of continuous evolution.

Previously [4], we defined a synchronization product between TDSHA which is presented in Appendix C where it is used to define the mapping used in our previous work on stochastic HYPE [4].

5 Mapping stochastic HYPE to TDSHA

We present now a mapping from stochastic HYPE to TDSHA which is based on the LTS obtained by the operational semantics. As mentioned before, this construction is different from the one defined previously in [4], in which we converted each subcomponent and each controller of stochastic HYPE to TDHSA, and then combined these TDSHA with a synchronization product. Essentially, the operational semantics takes care of this synchronization, and avoids the complications of the TDSHA product construction.

The mapping from LTS to TDSHA is quite straightforward: modes are given by the derivative set of the controlled system *ConSys*, namely by the set of reachable configurations, continuous transitions are extracted from the state in each configuration, and instantaneous and stochastic transitions correspond to the LTS transitions. Multiplicities of transitions labelled with stochastic events have to be respected to properly account for the quantitative behaviour of stochastic events, but as mentioned previously, we ignore multiplicity of instantaneous transitions, choosing to have a single transition with weight one. We assume that multiplicity of elements in a multiset is always respected in the definitions below.

Consider a stochastic HYPE model $(\mathit{ConSys}, \mathscr{V}, \mathit{IN}, \mathit{IT}, \mathscr{E}_d, \mathscr{E}_s, \mathscr{A}, ec, iv, \mathit{EC}, \mathit{ID})$, and let $\langle P_0, \sigma_0 \rangle$ be the configuration in the LTS obtained from *ConSys* after the execution of the initial event, $\underline{\text{init}}$, namely $\langle \mathit{ConSys}, \sigma \rangle \xrightarrow{\underline{\text{init}}} \langle P_0, \sigma_0 \rangle$. In the case of a well-defined HYPE model σ_0 does not depend on σ, as shown by [20] and this applies to stochastic HYPE models as well. We also let $\mathbf{v_0}$ be the value of variables $\mathscr{V}$ after initialisation.

In the following, we will refer to the activation condition and the reset of an event $\mathrm{a} \in \mathscr{E}$ by $\mathit{act}(\mathrm{a})$ and $\mathit{res}(\mathrm{a})$, respectively; hence $ec(\mathrm{a}) = (\mathit{act}(\mathrm{a}), \mathit{res}(\mathrm{a}))$. Recall that, for an instantaneous event $\underline{\mathrm{a}}$, $\mathit{act}(\underline{\mathrm{a}})$ is a boolean formula while, for a stochastic event $\overline{\mathrm{a}}$, $\mathit{act}(\overline{\mathrm{a}})$ is a function taking values in the positive reals. Given a state $\sigma = \{\iota_i \mapsto (r_i, I_i) \mid i = 1, \ldots k\}$, we indicate with $\mathbf{1}_{iv(\iota_i)}$ the k-vector equal to 1 in the coordinate corresponding to variable $iv(\iota_i)$ and zero elsewhere.

The reset $\mathit{res}(\mathrm{a})$ is defined as the conjunction of atoms of the form $V \sim \theta(\mathscr{V})$. To obtain a reset in the TDHSA, each distribution $\mathscr{X}_i$ mentioned in $\theta(\mathscr{V})$ must be associated with a distinct variable W_i, then the reset function for a transition labelled with a is $R_{\mathrm{a}}(\mathscr{V}, \mathbf{W})$ such that each random variables W_i is drawn from its associated distribution.

The mapping now defined differs from the approach we took for standard HYPE, since it requires stochastic transitions.

Definition 7. *Let* $\mathscr{H} = (ConSys, \mathscr{V}, IN, IT, \mathscr{E}_d, \mathscr{E}_s, \mathscr{A}, ec, iv, EC, ID)$ *be a stochastic HYPE model. The TDSHA* $\mathscr{T}(\mathscr{H}) = (Q, \mathbf{X}, \mathscr{TC}, \mathscr{TD}, \mathscr{TS}, \text{init}, \mathscr{E})$ *associated with* $\mathscr{H}$ *is defined by:*

- *the set of discrete modes is the derivative set of ConSys:* $Q = ds(ConSys)$*;*
- *the set of continuous variables is* $\mathbf{X} = \mathscr{V}$*;*
- *the initial state is* $\text{init} = (\langle P_0, \sigma_0 \rangle, R_{\underline{\text{init}}})$*;*
- *the set of events is* $\mathscr{E} = \mathscr{E}_d \cup \mathscr{E}_s$*;*
- *the multiset of continuous transitions* $\mathscr{TC}$ *is constructed as follows: with each* $\langle P, \sigma \rangle \in Q$*,* $\sigma = \{\iota_i \mapsto (r_i, I_i) \mid i = 1, \ldots k\}$*, we associate the continuous transitions* $(\langle P, \sigma \rangle, \mathbf{1}_{iv(\iota_i)}, r_i.[\![I_i]\!])$*,* $i = 1, \ldots k$*;*
- *the set of instantaneous transitions* $\mathscr{TD}$ *is obtained as follows: with each* $\langle P_1, \sigma_1 \rangle \xrightarrow{\underline{a}} \langle P_2, \sigma_2 \rangle$ *we associate the instantaneous transition* $(\langle P_1, \sigma_1 \rangle, \langle P_2, \sigma_2 \rangle, act(\underline{a}), R_{\underline{a}}, p, \underline{a})$ *with weight* $p = 1$*. We ignore multiplicity in the LTS.*
- *the multiset of stochastic transitions* $\mathscr{TS}$ *is constructed as follows: with each* $\langle P_1, \sigma_1 \rangle \xrightarrow{\overline{a}} \langle P_2, \sigma_2 \rangle$ *taking into account multiplicity in the LTS we associate the stochastic transition* $(\langle P_1, \sigma_1 \rangle, \langle P_2, \sigma_2 \rangle, true, R_{\overline{a}}, act(\overline{a}), \overline{a})$ *with guard true and rate function* $act(\overline{a})$ *and thus preserve multiplicity.*

This definition gives the same TDSHA as that in the earlier work on stochastic HYPE [4] which used a product construction (the compositional mapping); however, we prefer to work first with a labelled transition system before mapping to a TDHSA as it allows us to define bisimulation at the level of stochastic HYPE models. We can prove that the semantics defined above and those presented in [4] map a stochastic HYPE model to the same TDSHA and this theorem and its proof can be found in Appendix C.

A single trajectory of the node example from Section 2 is given in Figure 3. The proportion of time that the input connection is up and the rate of input are both higher than that of the output node, and this can be seen in the fact that the slopes for input are steeper, and the buffer is hitting its maximum capacity frequently. The failure that causes packets to be dropped does, paradoxically, enable the buffer to accept more packets when there is a connection.

We next consider some technical issues of model behaviour, as well a syntactic abbreviation for events with non-exponential durations. After this, the example of an assemble line shows the versatility of stochastic HYPE and this is followed by the definition of an equivalence over stochastic HYPE models that provides reasoning about when two models have the same behaviour.

5.1 Well-behaved models

In the paper that defines HYPE [20], a notion of well-behavedness is introduced. This notion ensures that the model can never execute an infinite number of simultaneous instantaneous events (called instantaneous Zeno behaviour), hence ensuring that models can be simulated. We want stochastic HYPE models to be similarly well-behaved, so that models can be simulated and additionally so that the TDHSA that a stochastic HYPE model is mapped to, can be interpreted as piecewise deterministic Markov process (PDMP) [15]. The definition of PDMPs require that an instantaneous event is not immediately followed by another one, and if the behaviour of the stochastic HYPE model is such that in its TDHSA there are only finite sequences of simultaneous events, then each finite sequence can be mapped to a single event and thus satisfy the PDMP definition.

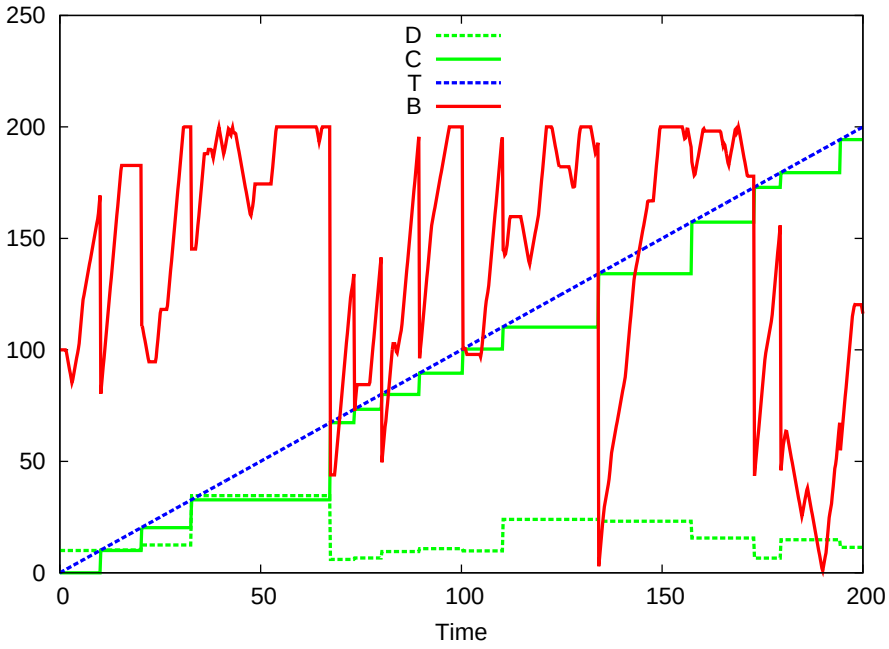

Figure 3: A single trace for the network node with $max_B = 200$, $k_{in}^{on} = 0.4$, $k_{in}^{off} = k_{out}^{off} = k_{out}^{on} = 0.2$, $r_{in} = 20$, $r_{out} = 10$, $\Delta = 2.5$, $\xi = 0.5$ and $b_0 = 100$. The variable B represents the quantity of packets in the buffer, T the time, C the variable that records the time, and D the duration.

Definition 8. *A stochastic HYPE model P is well-behaved if it has a finite number of finite sequences of simultaneous instantaneous events and these sequences are independent of the initial state of the system.*

To ensure well-behavedness for a HYPE model (as defined in [20], so without stochastic events), it is sufficient to show that the instantaneous activation graph, or I-graph, of a HYPE model is acyclic. This graph is constructed by considering the instantaneous transitions of the labelled transition systems obtained from a HYPE model. The I-graph gives an overapproximation of a HYPE model's behaviour. If there are no cycles then the model is well-behaved, but the I-graph of a well-behaved model is not necessarily acyclic.

Considering stochastic HYPE models, the addition of random resets has a minor effect on the construction on the I-graph in that for each event the possible set of values after a reset must take into account the support of any distribution involved in the reset, which can lead to additional overapproximation. The extension to stochastic events does not alter this construction and all results from [20] hold for stochastic HYPE models. The addition of a stochastic event between two instantaneous events, or alternatively, the modification of an instantaneous event to a stochastic one may break the cyclicity of an I-graph thus leading to a well-behaved model. Hence, both Theorem 6.1 and Theorem 6.2 proved by [20] hold for stochastic HYPE, as well as Propositions 6.1 to 6.4. which describe specific conditions on controllers that lead to well-behaved models. A new proposition can be proved in the stochastic setting.

Proposition 1. *Let P be a stochastic HYPE model with $Con \stackrel{\text{def}}{=} \mathrm{a}_1 \ldots \mathrm{a}_n.Con$. If there exists i such that $\bar{\mathrm{a}}_i$ is a stochastic event then P is well-behaved.*

Proof sketch. Since *Con* cycles through n events, if one of these events is not instantaneous and therefore has a duration then it is not possible for there to be an infinite sequence of instantaneous events. □

This proposition cannot be applied to the *Buffer* example, because the controllers are not cyclical, but a similar argument can be made. For the one instantaneous event in each controller (full or empty), a stochastic event must occur before the instantaneous event can reoccur, hence preventing the unwanted behaviour. Considering the two controllers in cooperation, they have disjoint events and hence the overall controller consists of interleavings of these events. It is not possible for a sequence to occur consisting only of full and empty since there must be interleaving stochastic events. Hence *Buffer* is well-behaved. This can also be proved using Proposition 6.4 from [20] since neither full nor empty activate each other, so one cannot immediately proceed the other. Considering the controller Con_{fail}, the only way in which multiple fail events can happen simultaneously is if the random value for the next duration is repeatedly zero. The probability of this is zero and hence the controller is well-behaved. Proposition 6.4 can be used again to show that the composition of all three controllers is well-behaved.

5.2 Non-exponential durations

We also introduce some syntax that will allow us to write more compact models. Currently the duration of stochastic events is specified by exponential distributions because this is a good match with TDSHA and PDMPs and more particularly because it avoids the need for residual clocks. However, we can allow a notation whereby any expression involving random variables can appear as the first element of an event condition. This provides a way to express any random duration directly in an event condition. This will then be expanded to two events, and requires the introduction of two variables, one to record the current time and one to record the duration of the event. This introduces a specific timer to track how much time is left of a duration. To illustrate this, consider the following event.

$$ec(\underline{\text{fail}}) \quad = \quad (T = C+D,\, C' \sim T \wedge D' \sim \mathscr{N}(\Delta,\xi)) \wedge B' \sim B - \mathscr{U}(0,B))$$

This can be written as

$$ec(\overline{\text{fail}}) \quad = \quad (\mathscr{N}(\Delta,\xi), B' \sim B - \mathscr{U}(0,B))$$

If there is no Timer subcomponent with an influence affecting a time variable T, then these must also be added. Since an expression that contains no random variables is interpreted as the rate of an exponential distribution, the notation $\delta(p)$ where p contains no random variables will be used to denote a fixed-time duration of p time units.

We have now defined the dynamics of stochastic HYPE system as well as highlight how models can have desirable behaviour. We assume well-behavedness in the rest of the paper. We proceed with an example after which we consider how we can formally compare two systems in terms of their behaviour.

6 Example: a manufacturing system

The example considers automated machines for assembling together groups of identical items, for example, putting matches into matchboxes. A schematic of the system is presented in Figure 4. Each machine determines if there are sufficient items in the pool ($\underline{\text{check}}_i$), then it takes these n_i items. Since this action can vary in duration it is modelled as a stochastic event ($\overline{\text{remove}}_i$) with a exponentially distributed duration. The next step is assembly and the event $\underline{\text{assem}}_i$ indicates the end of this fixed duration process when the machine places m_i finished items onto an output conveyor belt. Only one machine is allowed to take items from the pool at a time and this is enforced by a controller.

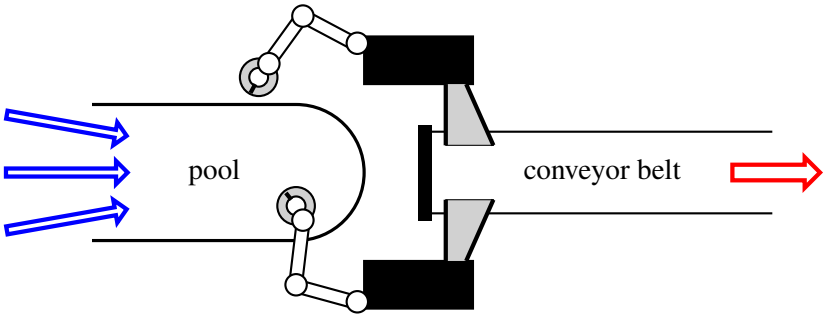

Figure 4: Schematic of the assembly system

There are three lines of items that feed into the pool. Completed items are removed from the conveyer belt. At the conveyer belt, there is an agent that stops the belt, inspects the items near to it, removes incorrectly assembled items and restarts the belt. If the beginning of the conveyer belt becomes congested and starts to overflow, the system moves to a failsafe state where everything stops.

Here, we treat both the input feed into the pool (P) and the output belt (B) as continuous. We also track the power consumption of each machine (W_i). The uncontrolled system is defined in Figure 5 and its controller are presented in Figure 6. *Con* consists of five controllers: one for each machine, a controller that controls access to the pool, a controller for inspection of assembled items, and a controller that determines the congestedness of the output belt and shuts down the whole system if necessary. Figure 7 provides a single run of the system.

We now consider a more complex scenario to show the potential of stochastic HYPE. This example involves the following cost optimisation problem. The manufacturer receives an order for K products that have to be produced within a certain deadline. Failure to meet the deadline will result in a penalty proportional to the delay. We assume that the assembly machines can be tuned by changing the batch size of a single assembly: the machine can take more items from the pool and put more assembled items on the belt. This has a cost in terms of assembly time and energy consumption. However, we assume that the production time increases as the square root of the batch size, while the energy increases quadratically. This models the fact that increasing the batch size reduces the production time but at an increased energy cost. Energy itself contributes to the total cost at a given price per unit. The goal of the manufacturing is to find the batch size that minimises the average cost, defined as the energy cost plus the penalty to miss the deadline. This design problem can be solved by exploring parameters in a feasible range and looking for the minimum value of the average cost, and results are reported in Figure 8 (taking the average over 1000 runs per point), suggesting an optimal batch size of 2. The model for this optimisation requires minor changes to the model in Figure 5 and these are easy to make because of the structured form of the model. Assembly times and the flows describing energy consumption are modified, and for the purposes of the example, the area where the manufactured items are placed is made much larger because the focus is on production, rather than problems related to overflow. No changes are required for the controllers.

7 Bisimulations for stochastic HYPE

We now consider behavioural equivalences for well-behaved stochastic HYPE models. First, we show that how the natural extension of the bisimulation defined for HYPE is not a useful definition when stochastic hybrid models are considered. Next, we define an equivalence that is more suited to capture the notion of same behaviour both stochastically and instantaneously.

$$
\begin{aligned}
\mathit{Assembler} &\stackrel{def}{=} \mathit{Sys} \underset{*}{\bowtie} \underline{\mathrm{init}}.\mathit{Con}_j \\
\\
\mathit{Sys} &\stackrel{def}{=} (\mathit{Feed}_1 \underset{*}{\bowtie} \mathit{Feed}_2 \underset{*}{\bowtie} \mathit{Feed}_3) \underset{*}{\bowtie} \mathit{Inspect} \underset{*}{\bowtie} \\
&\quad (\mathit{Timer}_1 \underset{*}{\bowtie} \mathit{Machine}_1(W_1)) \underset{*}{\bowtie} \\
&\quad (\mathit{Timer}_2 \underset{*}{\bowtie} \mathit{Machine}_2(W_2)) \\
\\
\mathit{Feed}_i &\stackrel{def}{=} \underline{\mathrm{init}} : (p_i, \mathit{arrivals}_i, \mathit{const}).\mathit{Feed}_i + \\
&\quad \underline{\mathrm{overflow}} : (p_i, 0, \mathit{const}).\mathit{Feed}_i \\
\\
\mathit{Inspect} &\stackrel{def}{=} \underline{\mathrm{init}} : (b, -\mathit{departures}, \mathit{const}).\mathit{Inspect} + \\
&\quad \overline{\mathrm{scan}} : (b, -\mathit{departures}, \mathit{const}).\mathit{Inspect} \\
&\quad \overline{\mathrm{resume}} : (b, 0, \mathit{const}).\mathit{Inspect} \\
&\quad \underline{\mathrm{overflow}} : (b, 0, \mathit{const}).\mathit{Inspect} \\
\\
\mathit{Machine}_i(W_i) &\stackrel{def}{=} \underline{\mathrm{init}} : (w_i, wa_i, \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) + \\
&\quad \underline{\mathrm{check}_i} : (w_i, 0, \mathit{const}).\mathit{Machine}_i(W_i) + \\
&\quad \overline{\mathrm{remove}_i} : (w_i, wt_i, \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) + \\
&\quad \underline{\mathrm{assem}_i} : (w_i, wa_i, \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) \\
&\quad \underline{\mathrm{overflow}_i} : (w_i, 0, \mathit{linear}(W_i)).\mathit{Machine}_i(W_i) \\
\\
\mathit{Timer}_i &\stackrel{def}{=} \underline{\mathrm{init}} : (t_1, 0, \mathit{const}).\mathit{Timer}_i + \\
&\quad \underline{\mathrm{remove}_i} : (t_1, 1, \mathit{const}).\mathit{Timer}_i + \\
&\quad \underline{\mathrm{assem}_i} : (t_1, 0, \mathit{const}).\mathit{Timer}_i \\
\\
\mathit{iv}(p_i) &= P \quad \mathit{iv}(t_i) = T_i \quad \mathit{iv}(w_i) = W_i \quad \mathit{iv}(b) = B \\
\\
\mathit{ec}(\underline{\mathrm{init}}) &= (\mathit{true},\ P' \sim P_0 \wedge T_1' \sim 0 \wedge T_2' \sim 0 \wedge B' \sim B_0) \\
\mathit{ec}(\underline{\mathrm{overflow}}) &= (B \geq B_f,\ \mathit{true}) \\
\mathit{ec}(\underline{\mathrm{check}_i}) &= (P \geq n_i,\ \mathit{true}) \\
\mathit{ec}(\underline{\mathrm{assem}_i}) &= (T_i \geq \mathit{atime}_i,\ B' \sim B + m_i) \\
\\
\mathit{ec}(\overline{\mathrm{remove}_i}) &= (\mathit{ttime}_i,\ P' \sim P - n_i \wedge T_i' \sim 0) \\
\mathit{ec}(\overline{\mathrm{scan}}) &= (\mathit{etime},\ B' \sim B - \mathit{min}(B, \Gamma(S_c, S_h))) \\
\mathit{ec}(\overline{\mathrm{resume}}) &= (\mathscr{F}(\mathit{rtime}),\ \mathit{true}) \\
\\
[\![\mathit{const}]\!] &= 1 \qquad [\![\mathit{linear}(X)]\!] = X
\end{aligned}
$$

Figure 5: Model for assembler with two machines (controller omitted)

$$
\begin{array}{rcl}
Con & \stackrel{def}{=} & ((C_1 \parallel C_2) \bowtie_{*} C_m) \parallel C_e \parallel C_f \\
\\
C_i & \stackrel{def}{=} & \underline{\text{check}}_i.C_i' \\
C_i' & \stackrel{def}{=} & \overline{\text{remove}}_i.C_i'' \\
C_i'' & \stackrel{def}{=} & \underline{\text{assem}}_i.C_i \\
\\
C_m & \stackrel{def}{=} & \underline{\text{check}}_1.C_m' + \underline{\text{check}}_2.C_m'' \\
C_m' & \stackrel{def}{=} & \overline{\text{remove}}_1.C_m \\
C_m'' & \stackrel{def}{=} & \overline{\text{remove}}_2.C_m \\
\\
C_e & \stackrel{def}{=} & \overline{\text{scan}}.\overline{\text{resume}}.C_e \\
\\
C_f & \stackrel{def}{=} & \underline{\text{overflow}}.0
\end{array}
$$

Figure 6: Controller for the assembly system

7.1 System bisimulation

We have previously defined system bisimulation for HYPE [20] and shown that it is the same as ic-bisimulation [12, 3]. The definition of system bisimulation along these lines for stochastic HYPE only requires the modification of the labels on the transitions so that they can either be stochastic or instantaneous.

Definition 9. *A relation $B \subseteq \mathscr{C} \times \mathscr{C}$ is a* system bisimulation *if for all $(P,Q) \in B$, for all* $\mathrm{a} \in \mathscr{E}_d \cup \mathscr{E}_s$*, for all $\sigma \in \mathscr{S}$ whenever*

1. $\langle P, \sigma \rangle \xrightarrow{\mathrm{a}} \langle P', \sigma' \rangle$*, there exists* $\langle Q', \sigma' \rangle$ *with* $\langle Q, \sigma \rangle \xrightarrow{\mathrm{a}} \langle Q', \sigma' \rangle$*,* $(P', Q') \in B$*.*
2. $\langle Q, \sigma \rangle \xrightarrow{\mathrm{a}} \langle Q', \sigma' \rangle$*, there exists* $\langle P', \sigma' \rangle$ *with* $\langle P, \sigma \rangle \xrightarrow{\mathrm{a}} \langle P', \sigma' \rangle$*,* $(P', Q') \in B$*.*

P and Q are system bisimilar*, $P \sim_s Q$ if they are in a system bisimulation.*

The results that held for HYPE also hold for stochastic HYPE, as dealing with stochastic events in the proofs is straightforward. Hence, system bisimulation is a congruence for Prefix, Choice and Parallel, and if two uncontrolled systems have the same set of prefixes, and are put in cooperation with the same controller, then the two controlled systems are bisimilar. System bisimulation can be lifted to the model level and congruence can be shown for stochastic HYPE model product. As these are all straightforward modifications of existing proofs, we omit presenting them formally for reasons of space. The reader is referred to [20] for further details.

System bisimulation is a static bisimulation in the sense that it does not consider the detailed behaviour of the model. It considers which events can occur, matches on them and also matches strictly on state. However, for stochastic events it requires exact matching of rates, rather than of the overall rate to processes with the same behaviour. In the next section, we consider a less strict and more useful equivalence.

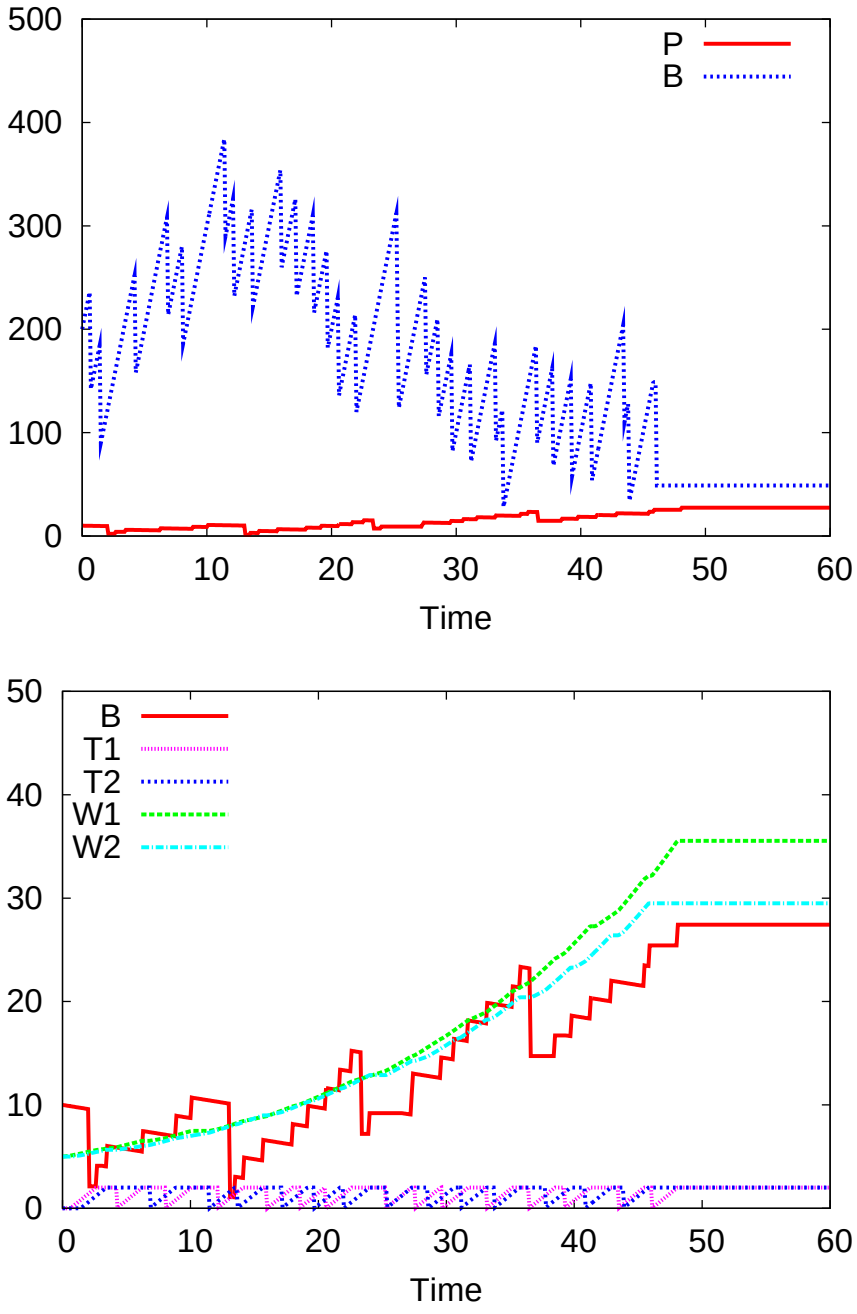

Figure 7: A single trace for the assembly system showing variables P and B (top) and variables B, T_i and W_i (bottom) using the parameters $arrivals_i = 20$, $departures = 0.2$, $atime_i = 2$, $ttime = 0.8$, $n_i = 100$, $m_i = 2$, $wt_i = 0.03$, $wa_i = 0.05$, $etime = 2$, $rtime = 20$, $S_c = 4$ $S_h = 0.5$ and $B_f = 25$

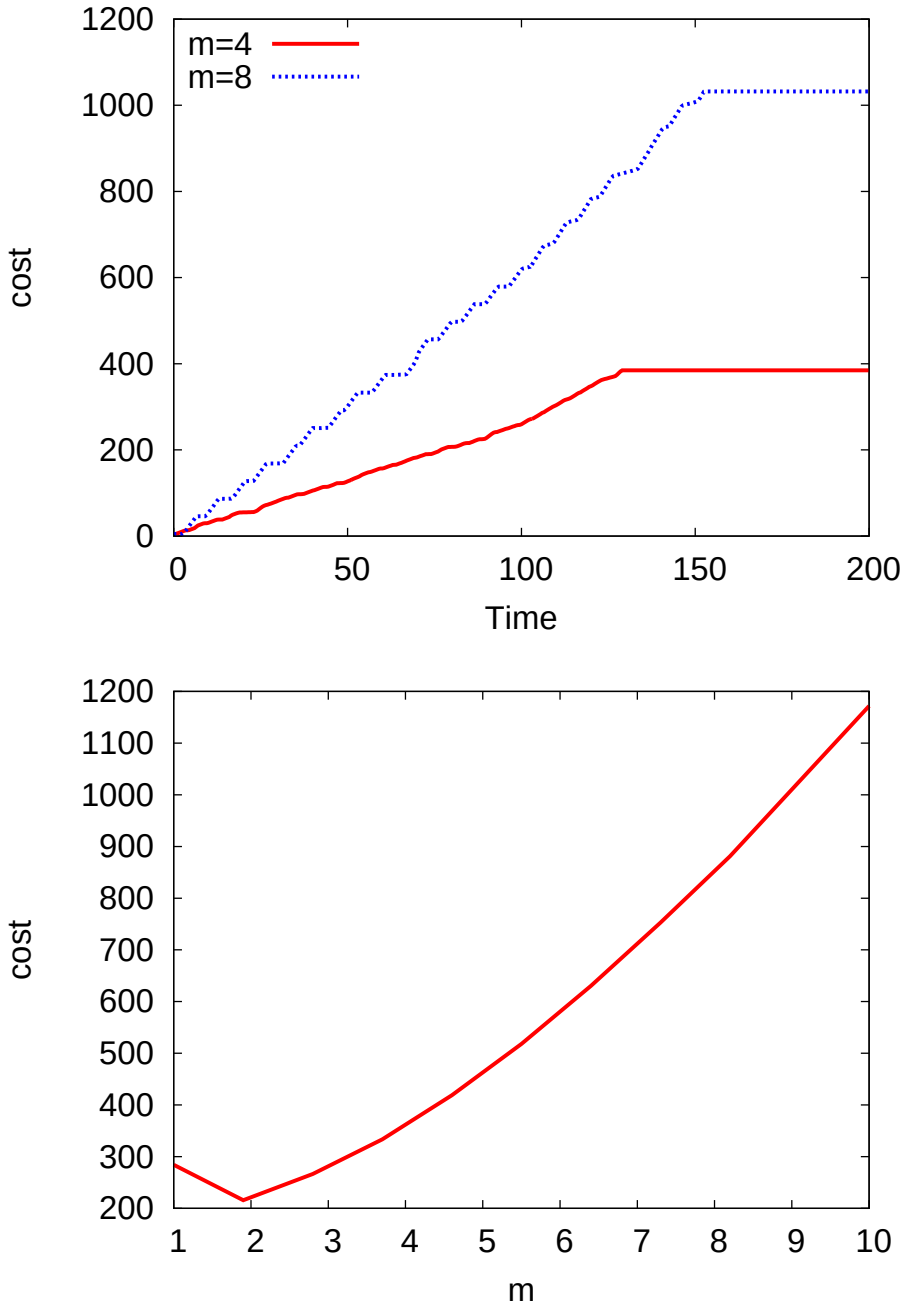

Figure 8: Energy plus penalty cost for various values of batch size m, assuming both machines are configured to the same value. Two trajectories for different values of m (top) and average cost for a range of values for m over 1000 trajectories (bottom). Energy cost per unit of energy is 0.5, while penalty cost is 2 per time unit. Energy rate consumption in the assembly phase is equal to $1/3 * m^2 + 2/3$, while the duration of the assembly equals $\alpha * (\sqrt{m} - 1) + 1.5$, for $\alpha = 1/(2 * sqrt(2) - 2)$. The number of items taken from the pool is $50m$. We assume the order is of 100 items with a deadline of 100 time units.

7.2 An equivalence based on stochastic bisimulation

We now consider a bisimulation that takes conditions on dynamics into account and relaxes the strict matching on states. Furthermore, we wish to relax the requirements for matching of stochastic transitions, and move to a definition that is similar to stochastic bisimulation [24, 9] where the combined rate to an equivalence class for each action is considered. For this, we require additional definitions. The sum in the definition is taken over a multiset.

Definition 10. *For the transitions* $\langle P_1,\sigma_1\rangle \xrightarrow{\underline{a}} \langle P_2,\sigma_2\rangle$ *let* $mult(\langle P_1,\sigma_1\rangle \xrightarrow{\underline{a}} \langle P_2,\sigma_2\rangle)$ *be the number of such transitions.*

Definition 11. *Given a stochastic HYPE controlled system* $P = \Sigma \bowtie_* Con$*, the function* $r : \mathscr{F} \times \mathscr{E}_s \times \mathscr{C} \to \mathbb{R}^+$ *is defined as*

$$r(\langle P,\sigma\rangle,\bar{a},\Sigma \bowtie_* Con') = \begin{cases} act(\bar{a}) \cdot mult(\langle P,\sigma\rangle \xrightarrow{\bar{a}} \langle \Sigma \bowtie_* Con',\sigma'\rangle) & \text{if } Con \xrightarrow{\bar{a}} Con' \\ 0 & \text{otherwise} \end{cases}$$

Furthermore, this function can be extended to sets $C \subseteq \mathscr{C}$*,* $r(\langle P,\sigma\rangle,\bar{a},C) = \sum\{r(\langle P,\sigma\rangle,\bar{a},Q) \mid Q \in C\}$.

Definition 12. *Given* $\equiv$*, an equivalence relation over states, an equivalence relation* $B \subseteq \mathscr{C} \times \mathscr{C}$ *is a* stochastic system bisimulation with respect to $\equiv$ *if for all* $(P,Q) \in B$*, for all* $\sigma \equiv \tau$*, for all* $C \in (\mathscr{F}/B)/\equiv$*,*

1. *for all* $\underline{a} \in \mathscr{E}_d$*, whenever*
   (a) $\langle P,\sigma\rangle \xrightarrow{\underline{a}} \langle P',\sigma'\rangle \in C$*, there exists* $\langle Q',\tau'\rangle \in C$ *such that* $\langle Q,\tau\rangle \xrightarrow{\underline{a}} \langle Q',\tau'\rangle$.
   (b) $\langle Q,\tau\rangle \xrightarrow{\underline{a}} \langle Q',\tau'\rangle \in C$*, there exists* $\langle P',\sigma'\rangle \in C$ *such that* $\langle P,\sigma\rangle \xrightarrow{\underline{a}} \langle P',\sigma'\rangle$.
2. *for all* $\bar{a} \in \mathscr{E}_s$*,* $r(\langle P,\sigma\rangle,\bar{a},C) = r(\langle Q,\tau\rangle,\bar{a},C)$.

P and Q are stochastic system bisimilar with respect to $\equiv$*,* $P \sim_s^{\equiv} Q$ *if they are in a stochastic system bisimulation with respect to* $\equiv$*.*

Note that even in the case that $\equiv$ is equality, stochastic system bisimulation with respect to $\equiv$ is less strict than isomorphism over the instantaneous transitions. Next, we consider under which conditions $\sim_s^{\equiv}$ is a congruence. We need two definitions to characterise the interaction of $\equiv$ and functions used in the operational semantics.

Definition 13. *An equivalence* $\equiv$ *over states* is preserved by updates *if* $\sigma \equiv \tau$ *implies that* $\sigma[\iota \mapsto (r,I)] \equiv \tau[\iota \mapsto (r,I)]$.

Definition 14. *An equivalence relation* $\equiv$ *over states* is preserved by Γ *if* $\sigma_i \equiv \tau_i$ *for* $i = 1,2,3$ *implies that* $\Gamma(\sigma_1,\sigma_2,\sigma_3) \equiv \Gamma(\tau_1,\tau_2,\tau_3)$.

Theorem 1. $\sim_s^{\equiv}$ *is a congruence for Influence, Choice and Cooperation, if* $\equiv$ *is preserved by updates and* Γ.

Proof sketch. Please refer to Appendix A. □

This theorem describes the conditions on the equivalence over states for congruence. Both preservation by updates and by Γ are strong conditions, but as we will see later in this section, not always necessary.

Because of the specific form of well-defined stochastic HYPE models, congruence with respect to some operators is less important. It is not possible to obtain a well-defined stochastic HYPE model by applying Prefix with Influence or Choice to an existing well-defined stochastic HYPE model. However Cooperation and Prefix are used to construct a controlled system from an uncontrolled system and controller, hence congruence can be used to prove further results as in the next theorem.

Theorem 2. *Given an uncontrolled system* Σ *and two controllers such that* $Con_1 \sim_s^{\equiv} Con_2$ *and let* $\equiv$ *be preserved by* Γ *then* $\Sigma \bowtie_* \underline{\text{init}}.Con_1 \sim_s^{\equiv} \Sigma \bowtie_* \underline{\text{init}}.Con_2$

Proof sketch. By congruence. □

Next, we introduce a specific equivalence over states, and prove that it gives the same ODEs for models that are stochastically system bisimilar with respect to it.

Definition 15. *Two states are equivalent,* $\sigma \doteq \tau$*, when for all* $V \in \mathscr{V}$ *and* $f(\mathscr{W})$*,* $\mathsf{sum}(\sigma, V, f(\mathscr{W})) = \mathsf{sum}(\tau, V, f(\mathscr{W}))$ *where*

$$\mathsf{sum}(\sigma, V, f(\mathscr{W})) = \sum \{\!\{ r \mid iv(\iota) = V, \sigma(\iota) = (r, I(\mathscr{W})), f(\mathscr{W}) = [\![I(\mathscr{W})]\!] \}\!\}$$

This equivalence abstracts from individual influences by requiring that the sum of strengths for each variable and influence type is preserved. It is not preserved by updates or Γ; however, as will be seen in the example section, it still provides a useful equivalence. This is because, for certain models, it is the case that $\sigma \doteq \tau$ implies $\Gamma(\sigma, \sigma', \sigma'') \doteq \Gamma(\tau, \tau', \tau'')$ even though $\sigma' \not\doteq \tau'$ and $\sigma'' \not\doteq \tau''$. This can be achieved by imposing additive conditions on the rates in a model for specific events. To illustrate this, consider the subcomponents

$$\begin{aligned} A_i &\stackrel{def}{=} \underline{\text{a}} : (k_i, r_i, I).A_i + \underline{\text{init}} : (k_i, 0, I).A_i \\ B_i &\stackrel{def}{=} \underline{\text{a}} : (j_i, s_i, I).B_i + \underline{\text{init}} : (j_i, t, I).B_i \end{aligned}$$

with $iv(k_i) = X = iv(j_i)$. After the $\underline{\text{init}}$ event, we have states $\sigma_i = \{k_i \mapsto (0, I), j_i \mapsto (t, I)\}$ and therefore $\mathsf{sum}(\sigma_i, X, I) = t$. Clearly, $\sigma_1 \doteq \sigma_2$. However, after an $\underline{\text{a}}$ event, we have states $\tau_i = \{k_i \mapsto (r_i, I), j_i \mapsto (s_i, I)\}$ and $\mathsf{sum}(\tau_i, A_i \bowtie_* B_i, I) = r_i + s_i$. Hence $\tau_1 \doteq \tau_2$ if and only if $r_1 + s_1 = r_2 + s_2$. This does not require that $r_1 = r_2$ and $s_1 = s_2$ which are the conditions required for equivalent states for A_1 and B_1, and A_2 and B_2.

Next, we wish to lift stochastic system bisimulation with respect to an equivalence, from controlled system level to model level, both to consider congruence of model product and to impose conditions on the elements of the tuples.

Definition 16. *Let* $(P_i, \mathscr{V}, IN, IT, \mathscr{E}_d, \mathscr{E}_s, \mathscr{A}, ec, iv, EC, ID)$ *for* $i = 1, 2$ *be two stochastic HYPE models. They are* stochastic system bisimilar with respect to an equivalance $\equiv$ over states *(denoted* $P_1 \sim_{\mathbf{sm}}^{\equiv} P_2$*) if* $P_1 \sim_s^{\equiv} P_2$.

Let the notation P_σ denote the collection of ODEs for model P in state σ. The next results shows that models that are stochastic system bisimilar with respect to $\doteq$ have the same ODEs.

Theorem 3. *Given two stochastic HYPE models* $(P, \mathscr{V}, IN, IT, \mathscr{E}_d, \mathscr{E}_s, \mathscr{A}, ec, iv, EC, ID)$ *and* $(Q, \mathscr{V}, IN, IT, \mathscr{E}_d, \mathscr{E}_s, \mathscr{A}, ec, iv, EC, ID)$*, if* $P \sim_{\mathbf{sm}}^{\doteq} Q$*, then for every configuration* $\langle P', \sigma_1 \rangle \in \mathrm{ds}(P)$ *and* $\langle Q', \sigma_2 \rangle \in \mathrm{ds}(Q)$ *such that* $P' \sim_{\mathbf{sm}}^{\doteq} Q'$*,* $P_{\sigma_1} = Q_{\sigma_2}$.

Proof sketch. Please refer to Appendix B. □

We can also define bisimulation at the TDSHA level and relate the stochastic HYPE bisimulation to that of the bisimulation over TDSHA. In Appendix D, we show that two stochastic HYPE models that are stochastic system bisimilar with respect to $\doteq$ have TDSHAs that are bisimilar but the converse does not necessarily hold.

$$Con_D \stackrel{def}{=} D \parallel C_e \parallel C_f$$

$$\begin{aligned}
D &\stackrel{def}{=} \underline{\text{check}}_1.D_{1,1} + \underline{\text{check}}_2.D_{1,2} \\
D_{1,i} &\stackrel{def}{=} \overline{\text{remove}}_i.D_{2,i} \\
D_{2,i} &\stackrel{def}{=} \underline{\text{assem}}_i.D + \underline{\text{check}}_i.D_{3,i} \\
D_{3,i} &\stackrel{def}{=} \underline{\text{assem}}_i.D_{1,i+1} + \overline{\text{remove}}_i.D_4 \\
D_4 &\stackrel{def}{=} \underline{\text{assem}}_1.D_{2,2} + \underline{\text{assem}}_2.D_{2,1}
\end{aligned}$$

Figure 9: Controller for the assembly system with a single controller for the two assembly machines (addition is modulo 2)

8 Example revisited: equivalence

We can now consider equivalence in the context of the manufacturing system. We can define Con_D, a composite controller for two machines and access to the pool as given in Figure 9, to replace the controllers C_1, C_2 and C_m.

First, we show that $Sys \bowtie_* \underline{\text{init}}.Con \sim_s^{\equiv} Sys \bowtie_* \underline{\text{init}}.Con_D$ for a suitable $\equiv$. This can be done by Theorem 2 and requires that a suitable equivalence relation be identified over the two controllers. We can ignore states inf configurations since they are not affected by the events in the controller. The relation B that follows is a stochastic system bisimulation with respect to $=$, and illustrates that therefore $(C_1 \parallel C_2) \bowtie_* C_m) \sim_s^{=} D$ and by congruence $Con \sim_s^{=} Con_D$, since equality preserves Γ. Hence $Sys \bowtie_* \underline{\text{init}}.Con \sim_s^{=} Sys \bowtie_* \underline{\text{init}}.Con_D$.

$$\begin{aligned}
B = \{&((C_1 \parallel C_2) \bowtie_* C_m, D), ((C_1'' \parallel C_2'') \bowtie_* C_m, D_4), \\
&((C_1' \parallel C_2) \bowtie_* C_m', D_{1,1}), ((C_1 \parallel C_2') \bowtie_* C_m'', D_{1,2}), \\
&((C_1'' \parallel C_2) \bowtie_* C_m, D_{2,1}), ((C_1 \parallel C_2'') \bowtie_* C_m, D_{2,2}), \\
&((C_1'' \parallel C_2') \bowtie_* C_m, D_{3,1}), ((C_1' \parallel C_2'') \bowtie_* C_m, D_{3,2})\}
\end{aligned}$$

Figure 10 shows the average for these two assembly systems over 5000 simulations. The similarity between these averages suggest that our definition of bisimulation has captured the similarities between the two systems.

We note here that the stochastic HYPE system *Assembler* (which uses the five subcontrollers) is well-behaved, according to Definition 8. This holds because each component of the controller is of the form required by Proposition 1, and contains a stochastic event ($\overline{\text{remove}}_i$). To show that $Assembler_D$ (which uses three subcontrollers one of which is Con_D) is well-behaved, we can simply invoke bisimilarity of the two systems, noticing that well-behavedness, being a condition on sequences of events, is preserved by bisimulation.

As mentioned above, the subcontrollers $(C_1 \parallel C_2) \bowtie_* C_m$ and D both ensure that only one machine has access to the pool at a time. Another approach is to use only the controller $(C_1 \parallel C_2)$ and modify the event conditions for each machine. We add a new variable M and redefine some event conditions.

$$\begin{aligned} ec(\underline{\text{init}}) &= (true,\ P' \sim P_0 \wedge T_1' \sim 0 \wedge T_2' \sim 0 \wedge B' \sim B_0 \wedge M' \sim 0) \\ ec(\underline{\text{check}}_i) &= (P \geq n_i \wedge M = 0,\ M' \sim 1) \\ ec(\overline{\text{remove}}_i) &= (ttime_i,\ P' \sim P - n_i \wedge T_i' \sim 0 \wedge M' \sim 0) \\ ec(\underline{\text{assem}}_i) &= (T_i \geq atime_i,\ B' \sim B + m_i) \end{aligned}$$

This modifies the system from one where mutual exclusion is determined by explicit sequencing of actions in the controller to one where a semaphore is used. Since the definition of stochastic system bisimulation has a requirement for events conditions to be equal for each event, we can no longer directly use this. Instead we can reason about the behaviour of the controllers from each system and show that the beahviour are the same, taking into account the different event conditions. We can then use Theorem 2 to argue for the stochastic system bisimilarity of the two controlled systems.

We wish to show that $C_1 \parallel C_2$ with these event conditions has the same behaviour as $(C_1 \parallel C_2) \bowtie_* C_m$. The labelled transition system of $C_1 \parallel C_2$ has nine derivatives (including itself). Eight of these derivatives have the same derivatives in the labelled transition systems of $(C_1 \parallel C_2) \bowtie_* C_m$, if we drop the contribution by derivatives of C_m. The derivative that does not appear is $C_1' \parallel C_2'$ where each controller has performed an $\underline{\text{check}}_i$ action and can then perform a $\underline{\text{remove}}_i$ action. If we can show that the value of the variable M ensures that this derivative cannot occur, then it is possible to contruct an isomorphism between the two LTSs and also to conclude that the two systems (one using $(C_1 \parallel C_2) \bowtie_* C_m$ as the controller, and one using $C_1 \parallel C_2$ with additional modification to the new variable M) are stochastic system bisimilar with respect to equality. To see that the derivative $C_1' \parallel C_2'$ is not reachable from the initial state, consider that if the first machine has performed $\underline{\text{check}}_1$ then M now has value 1, and it is not possible for the second machine to perform $\underline{\text{check}}_1$ because it has the guard that requires M to be zero. M is only reset to zero when the event $\overline{\text{remove}}_1$ ends. Hence it is not possible for C_2 to become C_2' until C_1' has become C_1''. This ensures $C_1' \parallel C_2'$ cannot happen. A similar argument applies if the second machine executes $\underline{\text{check}}_2$ first. Thus the bisimilarity is established.

Next, we consider the use of the bisimulation $\sim_s^{\doteq}$ where by using $\doteq$ we require that flows in state have a weaker form of equivalence than equality. We illustrate this through allowing different arrival rates for the feeds into the pool. Let Sys_{a_1,a_2,a_3} be the system such that

$$\begin{aligned} Feed_1 &\stackrel{def}{=} \underline{\text{init}}:(p_1,a_1,const).Feed_1 + \underline{\text{full}}:(p_1,0,const).Feed_1 \\ Feed_2 &\stackrel{def}{=} \underline{\text{init}}:(p_2,a_2,const).Feed_2 + \underline{\text{full}}:(p_2,0,const).Feed_2 \\ Feed_3 &\stackrel{def}{=} \underline{\text{init}}:(p_3,a_3,const).Feed_3 + \underline{\text{full}}:(p_3,0,const).Feed_3 \end{aligned}$$

Then $Sys_{a_1,a_2,a_3} \bowtie_* \underline{\text{init}}.Con_1 \sim_s^{\doteq} Sys_{b_1,b_2,b_2} \bowtie_* \underline{\text{init}}.Con_1$ whenever $a_1+a_2+a_3 = b_1+b_2+b_3$. The ODE that describes the change in the amount of items in the pool is $dP/dt = a_1+a_2+a_3 = k$ and $\mathsf{sum}(\sigma,P,const) = \mathsf{sum}(\tau,P,const)$. Hence as long as the systems being compared have the same value for k, the behaviour will be bisimilar. Replacing the $Feed_i$ by a single subcomponent

$$Feed \stackrel{def}{=} \underline{\text{init}}:(p,a,const).Feed + \underline{\text{full}}:(p,0,const).Feed$$

with $iv(p) = P$, also provides a bisimilar model with respect to $\doteq$ as long as $a = k$.

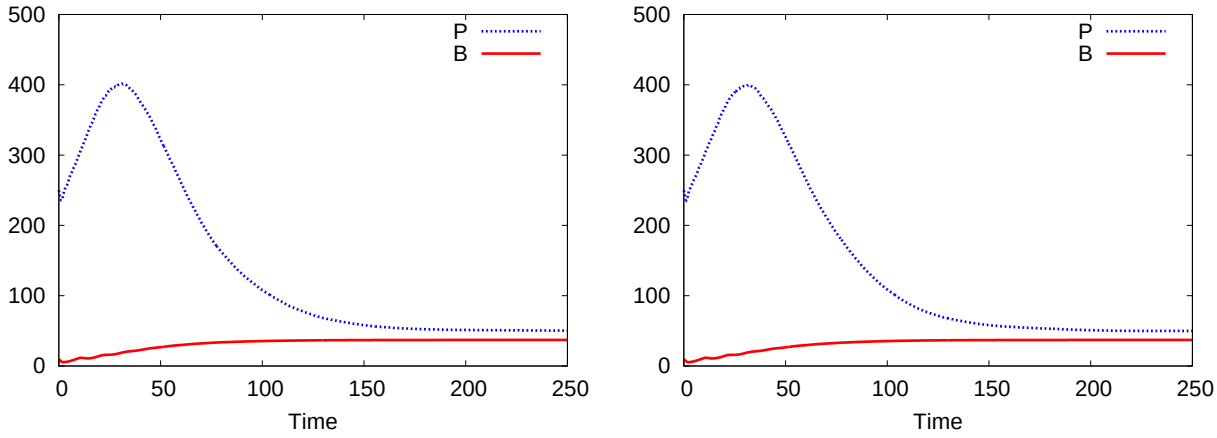

Figure 10: Average values for P and B for *Assembler* over 10000 simulations (left) and average values for P and B for $Assembler_D$ over 10000 simulations (right) using the same parameters as in Figure 7.

9 Related work

As described previously [20], HYPE takes a finer grained, less monolithic approach than the other process algebras for hybrid systems [3, 2, 30, 13] because it enables the modelling of individual flows. [26] compares these other process algebras based on the train gate controller example and for each of these process algebras, complete ODEs appear in the syntactic description of the system. Likewise, hybrid automata [23] require explicit definition of ODEs, and are less compositional than HYPE since the product of hybrid automata, requires disjoint variable sets. By contrast, HYPE product [20] does not require this and hence product construction allows shared variables in each component and hence richer behaviour through interaction. Other compositional hybrid formalisms such as CHARON [1], SHIFT [16], and HyCharts [22] do not map directly to hybrid automata, compared to HYPE. Hybrid action systems [29], Hybrid Interacting Maude [17] and bond graphs [28, 11] are other approaches that take a less monolithic approach.

Stochastic HYPE retains the fine grained approach to modelling flows, and the more expressive product construction, as well as adding the ability to model stochastic events. To the best of our knowledge, it is the only process algebra with these modelling capabilities.

Other formalisms for stochastic hybrid modelling include TDSHA [6, 7], PDMP [15], stochastic hybrid automata (SHA) [10] and the recent stochastic extension of UPPAAL [14]. Stochastic HYPE provides a language for reasoning about some of these formalisms since these are the semantic objects described by the stochastic HYPE syntax. TDSHA were developed to provide a transition-focussed approach to PDMP and hence providing a more consistent manner of treating the three different types of behaviour considered: continuous, stochastic and instantaneous behaviour.

Research into making stochastic hybrid systems modelling compositional has been considered [31], where Communicating Piecewise Deterministic Markov Processes (CPDP) are introduced. This is an automata-based formalism which models a system as interacting automata. Their chosen level of abstraction is somewhat lower level than ours, comparable with TDSHA. In CPDP, as in stochastic HYPE, instantaneous transitions may be triggered either by conditions of the continuous variables (boundary-hit transitions) or by the expiration of a stochastically determined delay (Markov transitions). Interaction

between automata is based on *one-way* synchronisation: in each interaction one partner is active while the other is passive. In stochastic HYPE, instead, all components may be regarded as active with respect to each transition in which they participate, as activation conditions are specified uniquely in the model. Components participating in a discrete transition are determined by the construction of the stochastic HYPE model, where the synchronisation set L in $\bowtie_L$ specifies which actions must be shared.

The synchronization mechanics of CPDP has been extended by [33], introducing an operator which exploits all possible interactions of active and passive actions. [32] define a notion of bisimulation for both PDMPs and CPDPs and show that if CPDPs are bisimilar then they give rise to bisimilar PDMPs. Furthermore the equivalence relation is a congruence with respect to the composition operator of CPDPs. The definition of bisimilarity presented in Appendix D is based on these definitions for PDMPs and CPDPs.

Stochastic HYPE has been used to model various systems, including an orbiter [4], a stochastic version of the train gate [20] and opportunistic networks [5]. More recently, a large scale model of ZebraNet [25], a wildlife monitoring project where individual zebra are nodes in an opportunistic network, has been developed which includes 2-dimensional animal movement, animal behaviour, and different opportunistic network protocols [18]. Stochastic HYPE has also been used to provide hybrid semantics for the biological process algebra Bio-PEPA [19] where reactions are treated stochastically or deterministically depending on species quantities or reaction rates.

10 Conclusion

In this paper we have presented an extension of the hybrid process algebra HYPE, in which non-urgent events fire at exponentially distributed random times. Although syntactically the modification that this entails is minimal with respect to the original version of HYPE, the semantics of the language is considerably enriched and quantified analysis of the modelled behaviour becomes available purely based on the model. The stochastic hybrid systems obtained from stochastic HYPE models fall in the class of Piecewise Deterministic Markov Processes (PDMP) [15]. Here we have shown how such a semantics can be derived through the intermediary of Transition-Driven Stochastic Hybrid Automata (TDSHA) [6]. The mapping that we present from stochastic HYPE to TDSHA differs from that originally presented in [4], as we work at the level of the labelled transition system generated by the operational semantics. Nevertheless we show that the obtained TDSHA in each case is equivalent (in Appendix C). We have discussed a number of bisumulation equivalences for stochastic HYPE with a particular focus on notions that are pragmatic and coincide with intuitive ideas of when behaviours coincide. Furthermore we have illustrated these with a case study based on an assembly line.

Acknowledgments: This work is partially supported by the EU project QUANTICOL, 600708.

References

[1] R. Alur, R. Grosu, I. Lee & O. Sokolsky (2006): *Compositional modeling and refinement for hierarchical hybrid systems*. Journal of Logic and Algebraic Programming 68, pp. 105–128.

[2] D.A. van Beek, K.L. Man, M.A. Reniers, J.E. Rooda & R.R.H. Schiffelers (2006): *Syntax and consistent equation semantics of hybrid χ*. Journal of Logic and Algebraic Programming 68, pp. 129–210.

[3] J.A. Bergstra & C.A. Middelburg (2005): *Process algebra for hybrid systems*. Theoretical Computer Science 335, pp. 215–280.

[4] L. Bortolussi, V. Galpin & J. Hillston (2011): *HYPE with stochastic events*. In: *Proceedings of the Ninth Workshop on Quantitative Aspects of Programming Languages (QAPL 2011)*, EPTCS 57, Saarbrücken, pp. 120–133.

[5] L. Bortolussi, V. Galpin & J. Hillston (2012): *Hybrid performance modelling of opportunistic networks*. In: *Proceedings of the Tenth Workshop on Quantitative Aspects of Programming Languages (QAPL 2012)*, EPTCS 85, Tallinn, pp. 106–121.

[6] L. Bortolussi & A. Policriti (2010): *Hybrid Dynamics of Stochastic Programs*. *Theoretical Computer Science* 411, pp. 2052–2077.

[7] L. Bortolussi & A. Policriti (2013): *(Hybrid) Automata and (Stochastic) Programs: The hybrid automata lattice of a stochastic program*. *Journal of Logic and Computation* 23, pp. 761–798.

[8] Luca Bortolussi (2012): *Hybrid Behaviour of Markov Population Models*. *CoRR* abs/1211.1643. Available at `http://arxiv.org/abs/1211.1643`.

[9] P. Buchholz (1995): *A notion of equivalence for stochastic Petri nets*. In: *Proceedings of the 16th International Conference on the Application and Theory of Petri Nets 1995*, pp. 161–180, doi:10.1007/3-540-60029-9_39.

[10] M.L. Bujorianu & J. Lygeros (2004): *General Stochastic Hybrid Systems: Modeling and Optimal Control*. In: *Proceedings of the 43rd IEEE CDC 2004*, pp. 1872–1877.

[11] P.J.L. Cuijpers, J.F. Broenink & P.J. Mosterman (2008): *Constitutive Hybrid Processes: a Process-Algebraic Semantics for Hybrid Bond Graphs*. *SIMULATION* 8, pp. 339–358.

[12] P.J.L. Cuijpers & M.A. Reniers (2003): *Hybrid Process Algebra*. Computer Science Reports CSR 03-07, Department of Computer Science, Eindhoven Technical University.

[13] P.J.L. Cuijpers & M.A. Reniers (2005): *Hybrid process algebra*. *Journal of Logic and Algebraic Programming* 62, pp. 191–245.

[14] A. David, D. Du, K.G. Larsen, A. Legay, M. Mikučionis, D.B. Poulsen & S. Sedwards (2012): *Statistical Model Checking for Stochastic Hybrid Systems*. In: *Proceedings of the First International Workshop on Hybrid Systems and Biology*, EPTCS 92, pp. 122–136.

[15] M.H.A. Davis (1993): *Markov Models and Optimization*. Chapman & Hall.

[16] A. Deshpande, A. Göllü & P. Varaiya (1996): *SHIFT: A Formalism and a Programming Language for Dynamic Networks of Hybrid Automata*. In P.J. Antsaklis, W. Kohn, A. Nerode & S. Sastry, editors: *Proceedings of Hybrid Systems IV*, LNCS 1273, pp. 113–133.

[17] M. Fadlisyah, P.C. Ölveczky & E. Ábrahám (2011): *Object-Oriented Formal Modeling and Analysis of Interacting Hybrid Systems in HI-Maude*. In: *SEFM 2011*, LNCS 7041, pp. 415–430.

[18] C. Feng (2012): *Modelling opportunistic networks with HYPE*. MSc dissertation, School of Informatics, University of Edinburgh.

[19] V. Galpin (2013): *Hybrid semantics for Bio-PEPA*. Under review for journal publication.

[20] V. Galpin, L. Bortolussi & J. Hillston (2013): *HYPE: Hybrid modelling by composition of flows*. *Formal Aspects of Computing* 25, pp. 503–541. Available at `http://dx.doi.org/10.1007/s00165-011-0189-0`.

[21] V. Galpin, J. Hillston & L. Bortolussi (2008): *HYPE applied to the modelling of hybrid biological systems*. *Electronic Notes in Theoretical Computer Science* 218, pp. 33–51.

[22] R. Grosu & T. Stauner (2002): *Modular and Visual Specification of Hybrid Systems: An Introduction to HyCharts*. *Formal Methods in System Design* 21, pp. 5–38.

[23] T.A. Henzinger (1996): *The Theory of Hybrid Automata*. In: *LICS*, pp. 278–292.

[24] J. Hillston (1996): *A compositional approach to performance modelling*. Cambridge University Press.

[25] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh & Daniel Rubenstein (2002): *Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet*. *ACM SIGPLAN Notices* 37, pp. 96–107.

[26] U. Khadim (2006): *A comparative study of process algebras for hybrid systems*. Computer Science Report CSR 06-23, Technische Universiteit Eindhoven. `http://alexandria.tue.nl/extra1/wskrap/publichtml/200623.pdf`.

[27] U. Khadim (2008): *Process algebras for hybrid systems: Comparison and development*. Ph.D. thesis, IPA, Technische Universiteit Eindhoven.

[28] H.M. Paynter (1961): *Analysis and Design of Engineering Systems*. MIT Press.

[29] M. Rönkkö, A.P. Ravn & K. Sere (2003): *Hybrid action systems*. Theoretical Computer Science 290, pp. 937–973.

[30] W.C. Rounds & H. Song (2003): *The Φ-Calculus: A Language for Distributed Control of Reconfigurable Embedded Systems*. In O. Maler & A. Pnueli, editors: Proceedings of HSCC 2003, LNCS 2623, pp. 435–449.

[31] S. Strubbe, A.A. Julius & A. J. van der Schaft (2003): *Communicating Piecewise Deterministic Markov Processes*. In: Proceedings of ADHS 2003, pp. 349–354.

[32] S. Strubbe & A.J. van der Schaft (2005): *Bisimulation for Communicating Piecewise Deterministic Markov Processes (CPDPs)*. In: Proceedings of the 8th International Workshop on Hybrid Systems: Computation and Control (HSCC 2005), LNCS 3414, pp. 623–639.

[33] S. Strubbe & A.J. van der Schaft (2005): *Stochastic semantics for Communicating Piecewise Deterministic Markov Processes*. In: Proceedings of IEEE CDC 2005, pp. 6103–6108.

A Congruence

Theorem 4. *$\sim_s^{\equiv}$ is a congruence for Influence, Choice and Cooperation, if $\equiv$ is preserved by updates and Γ.*

Proof sketch. Let $P_1 \sim_s^{\equiv} P_2$.

Prefix with influence We have the transition $\langle \mathrm{a}:(\iota,r,I).P_1,\sigma\rangle \xrightarrow{\mathrm{a}} \langle P_1,\sigma[\iota\mapsto(r,I)]\rangle$ and likewise the transition $\langle \mathrm{a}:(\iota,r,I).P_2,\tau\rangle \xrightarrow{\mathrm{a}} \langle P_2,\tau[\iota\mapsto(r,I)]\rangle$. Letting $\sigma\equiv\tau$, and since $\equiv$ is preserved by updates, we know that $\sigma[\iota\mapsto(r,I)]\equiv\tau[\iota\mapsto(r,I)]$. Moreover, for all P, σ and C, $r(\langle\bar{\mathrm{a}}.P,\sigma\rangle,\bar{\mathrm{c}},C) = act(\bar{\mathrm{a}})$ if $\bar{\mathrm{a}}=\bar{\mathrm{c}}$, otherwise $r(\langle\bar{\mathrm{a}}.P,\sigma\rangle,\bar{\mathrm{c}},C)=0$. Hence we can conclude that $\mathrm{a}:(\iota,r,I).P_1 \sim_s^{\equiv} \mathrm{a}:(\iota,r,I).P_2$.

Prefix without influence We know $\langle \underline{\mathrm{a}}.P_1,\sigma\rangle \xrightarrow{\mathrm{a}} \langle P_1,\sigma\rangle$ and $\langle \underline{\mathrm{a}}.P_2,\tau\rangle \xrightarrow{\mathrm{a}} \langle P_2,\tau\rangle$ with $\sigma\equiv\tau$. Furthermore $r(\langle\bar{\mathrm{a}}.P,\sigma\rangle,\bar{\mathrm{c}},C) = act(\bar{\mathrm{a}})$ if $\bar{\mathrm{a}}=\bar{\mathrm{c}}$ (otherwise 0), for any P and σ. Since $P_1 \sim_s^{\equiv} P_2$, then we can conclude that $\mathrm{a}.P_1 \sim_s^{\equiv} \mathrm{a}.P_2$.

Choice First, if $\langle P_1,\sigma\rangle \xrightarrow{\mathrm{a}} \langle P_1',\sigma'\rangle$, then we have the transition $\langle P_1+Q,\sigma\rangle \xrightarrow{\mathrm{a}} \langle P_1',\sigma'\rangle$ and since $P_1 \sim_s^{\equiv} P_2$, $\langle P_2,\tau\rangle \xrightarrow{\mathrm{a}} \langle P_2',\tau'\rangle$ with $P_1' \sim_s^{\equiv} P_2'$ and $\sigma'\equiv\tau'$, and hence $\langle P_2+Q,\tau\rangle \xrightarrow{\mathrm{a}} \langle P_2',\tau'\rangle$ as required. Second, if $\langle Q,\sigma\rangle \xrightarrow{\mathrm{a}} \langle Q',\sigma''\rangle$ then also $\langle Q,\tau\rangle \xrightarrow{\mathrm{a}} \langle Q',\tau''\rangle$ for some $\tau''\equiv\sigma''$ and $\langle P_2+Q,\tau\rangle \xrightarrow{\mathrm{a}} \langle Q',\tau''\rangle$. For stochastic events, $r(\langle P_1+Q,\sigma\rangle,\bar{\mathrm{a}},C) = r(\langle P_1,\sigma\rangle,\bar{\mathrm{a}},C) + r(\langle Q,\sigma\rangle,\bar{\mathrm{a}},C)$. Since $P_1 \sim_s^{\equiv} P_2$, $r(\langle P_1,\sigma\rangle,\bar{\mathrm{a}},C) = r(\langle P_2,\tau\rangle,\bar{\mathrm{a}},C)$. Moreover $r(\langle Q,\sigma\rangle,\bar{\mathrm{a}},C) = r(\langle Q,\tau\rangle,\bar{\mathrm{a}},C)$ and hence, the conclusion is that $r(\langle P_1+Q,\sigma\rangle,\bar{\mathrm{a}},C) = r(\langle P_2+Q,\tau\rangle,\bar{\mathrm{a}},C)$.

Cooperation We need to show that $B = \{(P_1 \underset{L}{\bowtie} Q, P_2 \underset{L}{\bowtie} Q) | P_1 \sim_s^{\equiv} P_2\}$ is a system bisimulation with respect to $\equiv$. There are three cases to consider. First, if $\langle P_1,\sigma\rangle \xrightarrow{\mathrm{a}} \langle P_1',\sigma'\rangle$ with $\underline{\mathrm{a}} \notin L$ then $\langle P_1 \underset{L}{\bowtie} Q,\sigma\rangle \xrightarrow{\mathrm{a}} \langle P_1' \underset{L}{\bowtie} Q,\sigma'\rangle$. Since $P_1 \sim_s^{\equiv} P_2$, for τ such that $\sigma\equiv\tau$, $\langle P_2,\tau\rangle \xrightarrow{\mathrm{a}} \langle P_2',\tau'\rangle$ with $\sigma'\equiv\tau'$ and $P_1' \sim_s^{\equiv} P_2'$, and hence $\langle P_2 \underset{L}{\bowtie} Q,\tau\rangle \xrightarrow{\mathrm{a}} \langle P_2' \underset{L}{\bowtie} Q,\tau'\rangle$. Next, considering stochastic events, we need to show that $r(\langle P_1 \underset{L}{\bowtie} Q,\sigma\rangle,\bar{\mathrm{a}},C) = r(\langle P_2 \underset{L}{\bowtie} Q,\tau\rangle,\bar{\mathrm{a}},C)$ for all equivalence classes $C \in (\mathscr{F}/B)/\equiv$. Since $r(\langle P_1 \underset{L}{\bowtie} Q,\sigma\rangle,\bar{\mathrm{a}},C) = r(\langle P_1,\sigma\rangle,\bar{\mathrm{a}},C) + r(\langle Q,\sigma\rangle,\bar{\mathrm{a}},C)$, we have the result. The second case where $\underline{\mathrm{a}} \notin L$ and $\langle Q,\sigma\rangle \xrightarrow{\mathrm{a}} \langle Q',\sigma''\rangle$ is proved in a similar fashion.

Third, consider, $\underline{\mathrm{a}} \in L$ and $\langle P_1,\sigma\rangle \xrightarrow{\mathrm{a}} \langle P_1',\sigma'\rangle$ and $\langle Q,\sigma\rangle \xrightarrow{\mathrm{a}} \langle Q',\sigma''\rangle$, then we have the transition $\langle P_1 \underset{L}{\bowtie} Q,\sigma\rangle \xrightarrow{\mathrm{a}} \langle P_1' \underset{L}{\bowtie} Q',\Gamma(\sigma,\sigma',\sigma'')\rangle$. Since $P_1 \sim_s^{\equiv} P_2$, and letting $\sigma\equiv\tau$, then $\langle P_2,\tau\rangle \xrightarrow{\mathrm{a}} \langle P_2',\tau'\rangle$ with $P_1' \sim_s^{\equiv} P_2'$ and $\sigma'\equiv\tau'$. Also $\langle Q,\tau\rangle \xrightarrow{\mathrm{a}} \langle Q',\tau''\rangle$ with $\sigma''=\tau''$. Hence we have the transition $\langle P_2 \underset{L}{\bowtie} Q,\tau\rangle \xrightarrow{\mathrm{a}} \langle P_2' \underset{L}{\bowtie} Q',\Gamma(\tau,\tau',\tau'')\rangle$ with $\Gamma(\sigma,\sigma',\sigma'') \equiv \Gamma(\tau,\tau',\tau'')$ as $\equiv$ is preserved by Γ. For stochastic transitions $\bar{\mathrm{a}} \in L$, we know that $r(\langle P_1 \underset{L}{\bowtie} Q,\sigma\rangle,\bar{\mathrm{a}},C) = act(\bar{\mathrm{a}}) \cdot \sum\{\!\{ mult(\langle P_1,\sigma\rangle \xrightarrow{\bar{\mathrm{a}}} F) \mid F \in C \}\!\} \cdot \sum\{\!\{ mult(\langle Q,\sigma\rangle \xrightarrow{\bar{\mathrm{a}}} F) \mid F \in C \}\!\}$, ensuring preservation of multiplicities. Since $P_1 \sim_s^{\equiv} P_2$, $r(\langle P_1,\sigma\rangle,\bar{\mathrm{a}},C) = r(\langle P_2,\tau\rangle,\bar{\mathrm{a}},C) = act(\bar{\mathrm{a}}) \cdot \sum\{\!\{ mult(\langle P_2,\tau\rangle \xrightarrow{\bar{\mathrm{a}}} F) \mid F \in C \}\!\}$, and hence we know $r(\langle P_2 \underset{L}{\bowtie} Q,\tau\rangle,\bar{\mathrm{a}},C) = r(\langle P_2 \underset{L}{\bowtie} Q,\sigma\rangle,\bar{\mathrm{a}},C)$.

□

B Equivalence of the two semantics

Theorem 5. *Let $(ConSys, \mathscr{V}, IN, IT, \mathscr{E}_c, \mathscr{E}_s, \mathscr{A}, ec, iv, EC, ID)$ be a stochastic HYPE model then the TDSHA obtained via the operational semantics is the same as the TDSHA obtained by the compositional mapping [4] when only reachable modes are considered.*

Proof sketch. Let $\langle P',\sigma_1\rangle \in \mathrm{ds}(P)$ and $\langle Q',\sigma_2\rangle \in \mathrm{ds}(Q)$ with $P' \sim_{\mathbf{sm}}^{\doteq} Q'$ then we can conclude that $\sigma_1 \doteq \sigma_2$. We need to identify the ODEs associated with each configuration. Since both of these are modes in the respective TDSHAs, we can identify their respective multisets of continuous transitions.

If $\sigma_1 = \{\iota_i \mapsto (r_i, I_i) \mid i = 1,\ldots k\}$ then the multiset of continuous transitions associated with $\langle P', \sigma_1\rangle$ is $\{\!|\, (\langle P, \sigma_1\rangle, \mathbf{1}_{iv(\iota_i)}, r_i.[\![I_i]\!]) \mid i = 1,\ldots k \,|\!\}$. Hence, for a given V_j, we obtain the ODE

$$\frac{dV_j}{dt} = \sum_{i=1}^{k} \{\!|\, r_i [\![I_i(\mathscr{V})]\!] \;\big|\; iv(\iota_i) = V_j \,|\!\}$$

This can be written more generally as follows.

$$P_{\sigma_1} = \left\{ \frac{dV}{dt} = \sum \{\!|\, r[\![I(\mathscr{V})]\!] \;\big|\; iv(\iota) = V, \sigma_1(\iota) = (r, I(\mathscr{V})) \,|\!\} \;\middle|\; V \in \mathscr{V} \right\}$$

Similarly, for $\langle Q', \sigma_2\rangle$ we have the following ODEs.

$$Q_{\sigma_2} = \left\{ \frac{dV}{dt} = \sum \{\!|\, r[\![I(\mathscr{V})]\!] \;\big|\; iv(\iota) = V, \sigma_2(\iota) = (r, I(\mathscr{V})) \,|\!\} \;\middle|\; V \in \mathscr{V} \right\}$$

Since $\sigma_1 \doteq \sigma_2$, for each ι, whenever $\sigma_1(\iota) = (r_1, I_1(\mathscr{V}))$ and $\sigma_2(\iota) = (r_2, I_1(\mathscr{V}))$ then $[\![I_1(\mathscr{V}))]\!] = [\![I_2(\mathscr{V})]\!]$ and for each $V \in \mathscr{V}$, $I \in IN$, $\mathsf{sum}(\sigma_1, V, I(\mathscr{V})) = \mathsf{sum}(\sigma_2, V, I(\mathscr{V}))$. Consider

$$\begin{aligned}
\frac{dV}{dt} &= \sum \{\!|\, r[\![I(\mathscr{V})]\!] \;\big|\; iv(\iota) = V, \sigma_1(\iota) = (r, I(\mathscr{V})) \,|\!\} \\
&= \sum_{I(\mathscr{V})} [\![I(\mathscr{V})]\!].\sum \{\!|\; \big|\; iv(\iota) = V, \sigma_1(\iota) = (r, I(\mathscr{V})) \,|\!\} \\
&= \sum_{I(\mathscr{V})} [\![I(\mathscr{V})]\!].\mathsf{sum}(\sigma_1, V, I(\mathscr{V})) = \sum_{I(\mathscr{V})} [\![I(\mathscr{V})]\!].\mathsf{sum}(\sigma_2, V, I(\mathscr{V})) \\
&= \sum_{I(\mathscr{V})} [\![I(\mathscr{V})]\!] \sum \{\!|\, r \;\big|\; iv(\iota) = V, \sigma_2(\iota) = (r, I(\mathscr{V})) \,|\!\} \\
&= \sum \{\!|\, r[\![I(\mathscr{V})]\!] \;\big|\; iv(\iota) = V, \sigma_2(\iota) = (r, I(\mathscr{V})) \,|\!\}
\end{aligned}$$

since for each $I(\mathscr{V})$, the rates obtained from the states sum to the same value. Therefore $P_{\sigma_1} = Q_{\sigma_2}$. □

C Comparison of TDSHA mappings

To show that the mapping which defined the semantics for stochastic HYPE [4] is the same as the mapping from SOS semantics presented here, the definition of the product of two TDHSAs is required as well as the mapping definition. We will refer to the mapping in [4] as the *compositional mapping* denoted $\mathscr{T}$, and will use *SOS mapping*, denoted $\mathscr{T}_{SOS}$, for the mapping in this paper. The compositional mapping did not use random resets but we do so here.

For the product, we require a consistency definition relating to resets, to ensure that no resets clash by attempting to set the same variable to two different values. Hence, we say that two transitions δ_1, δ_2 (either both discrete or both stochastic) are *reset-compatible* if and only if $e_{\delta_1} \neq e_{\delta_2}$ or $r_{\delta_1} \wedge r_{\delta_2} \neq \mathit{false}$. For random resets, this requires that any random variable (that necessarily appears in both resets) is drawn from the same distribution. Two TDSHA are reset-compatible if and only if all their discrete or stochastic transitions are pairwise reset-compatible. A similar notion is required for the initial conditions: Two TDSHA are *init-compatible* if and only if, given initial conditions $\mathrm{init}_1 = (q_1^{init}, \mathrm{inp}_1)$ and $\mathrm{init}_2 = (q_2^{init}, \mathrm{inp}_2)$, then $\mathrm{inp}_1 \wedge \mathrm{inp}_2 \neq \mathit{false}$.

In the definition of TDSHA product which follows, the set of continuous transitions in a mode $q = (q_1, q_2)$ contains all continuous transitions of q_1 and all those of q_2. The set of instantaneous transitions

$\mathscr{TD}$ is the union of non-synchronized instantaneous transitions $\mathscr{TD}_{NS}$ and of synchronized ones $\mathscr{TD}_S$, and during synchronization, a conservative policy is applied by taking the conjunction of guards and resets, and by taking the minimum of weights. Similarly, the set of stochastic transitions is defined as $\mathscr{TS} = \mathscr{TS}_{NS} \cup \mathscr{TS}_S$. In the synchronization of stochastic transitions, we use the fact that the rate is the same for all transitions labelled by the same event, as required by the consistency condition.

Definition 17. *Let $\mathscr{T}_i = (Q_i, \mathbf{X_i}, \mathscr{TC}_i, \mathscr{TD}_i, \mathscr{TS}_i, \text{init}_i, \mathscr{E}_i)$, $i = 1,2$ be two reset-compatible and init-compatible TDSHA, and let $L \subseteq \mathscr{E}_1 \cap \mathscr{E}_2$ be the synchronization set. The L-product $\mathscr{T} = \mathscr{T}_1 \otimes_L \mathscr{T}_2 = (Q, \mathbf{X}, \mathscr{TC}, \mathscr{TD}, \mathscr{TS}, \text{init}, \mathscr{E})$ is defined by*

1. $Q = Q_1 \times Q_2$;
2. $\mathbf{X} = \mathbf{X_1} \cup \mathbf{X_2}$;
3. $\mathscr{E} = \mathscr{E}_1 \cup \mathscr{E}_2$;
4. $\text{init} = (q^{init}, R_{\text{inp}})$, *where* $q^{init} = (q_1^{init}, q_2^{init})$ *and* $R_{\text{inp}} = R_{\text{inp}_1} \wedge R_{\text{inp}_2}$.
5. $\mathscr{TC} = \{\!\!\{ ((q_1,q_2), \mathbf{s}, f) \mid q_1 \in Q_1, q_2 \in Q_2, (q_1, \mathbf{s}, f) \in \mathscr{TC}_1 \vee (q_2, \mathbf{s}, f) \in \mathscr{TC}_2 \}\!\!\}$.
6. $\mathscr{TD} = \mathscr{TD}_{NS} \cup \mathscr{TD}_S$ *where*
$$\begin{aligned}
\mathscr{TD}_{NS} &= \{((q_1,q_2),(q_1',q_2'),g,r,w,e) \mid \\
&\qquad (q_i,q_i',g,r,w,e) \in \mathscr{TD}_i \wedge q_j = q_j' \in Q_j \wedge i \neq j \wedge e \notin S\}, \\
\mathscr{TD}_S &= \{((q_1,q_2),(q_1',q_2'),g_1 \wedge g_2, r_1 \wedge r_2, \min\{w_1,w_2\}, e) \mid \\
&\qquad (q_1,q_1',g_1,r_1,w_1,e) \in \mathscr{TD}_1 \wedge (q_2,q_2',g_2,r_2,w_2,e) \in \mathscr{TD}_2 \wedge e \in S\}.
\end{aligned}$$
7. $\mathscr{TS} = \mathscr{TS}_{NS} \cup \mathscr{TS}_S$ *where*
$$\begin{aligned}
\mathscr{TS}_{NS} &= \{\!\!\{ ((q_1,q_2),(q_1',q_2'),g,r,f,e) \mid \\
&\qquad (q_i,q_i',g,r,f,e) \in \mathscr{TS}_i \wedge q_j = q_j' \in Q_j \wedge i \neq j \wedge e \notin S \}\!\!\}, \\
\mathscr{TS}_S &= \{\!\!\{ ((q_1,q_2),(q_1',q_2'),g_1 \wedge g_2, r_1 \wedge r_2, f, e) \mid \\
&\qquad (q_1,q_1',g_1,r_1,f,e) \in \mathscr{TS}_1 \wedge (q_2,q_2',g_2,r_2,f,e) \in \mathscr{TS}_2 \wedge e \in S \}\!\!\}.
\end{aligned}$$

C.1 Compositional mapping from HYPE to TDSHA

The mapping $\mathscr{T}$ that is now defined, works compositionally, by associating a TDSHA with each single subcomponent and with each piece of the controller, then taking their synchronized product according to the synchronization sets of the stochastic HYPE system. Guards, rates, and resets of discrete edges will be incorporated in the TDSHA of the controller, while continuous transitions will be extracted from the uncontrolled system. Consider a stochastic HYPE model $(ConSys, \mathscr{V}, IN, IT, \mathscr{E}_d, \mathscr{E}_s, \mathscr{A}, ec, iv, EC, ID)$ with $ConSys ::= \Sigma \underset{L}{\bowtie} \underline{\text{init}}.Con$.

To define the TDSHA of the uncontrolled system Σ, consider a subcomponent $S = \sum_{i=1}^{k} \text{a}_i : \alpha_i.S + \underline{\text{init}} : \alpha.S$. Each element of $\in (S)$ generates a mode in the TDSHA of S. Moreover, in each such mode, the only continuous transition will be the one that is derived from this influence. As for discrete edges, since response to all events is always enabled, there will be an outgoing transition for each event appearing in S in each mode of the associated TDSHA. The target state of the transition will be the mode corresponding to the influence following the event. Resets and guards will be set to *true*, as event conditions are associated with the controller. Rates of transitions derived from stochastic events $\bar{\text{a}} \in \mathscr{E}_s$ will be set to $act(\bar{\text{a}})$, as required by the consistency condition of TDSHA. Finally, weights will be set to 1, while the initial mode will be deduced from the $\underline{\text{init}}$ event. Formally, this is defined as follows.

Definition 18. *Let $S(\mathscr{V}) = \sum_{i=1}^{k} \text{a}_i : \alpha_i.S(\mathscr{V}) + \underline{\text{init}} : \alpha.S(\mathscr{V})$ be a subcomponent of the stochastic HYPE model $(ConSys, \mathscr{V}, IN, IT, \mathscr{E}_d, \mathscr{E}_s, \mathscr{A}, ec, iv, EC, ID)$. Then $\mathscr{T}(S) = (Q, \mathbf{X}, \mathscr{TC}, \mathscr{TD}, \mathscr{TS}, \text{init}, \mathscr{E})$, the TDHSA associated with S is defined by*

1. $Q = \{q_\alpha \mid \alpha \in \mathsf{is}(S)\}$; $\mathbf{X} = \mathscr{V}$; $\mathscr{E} = \mathscr{E}_d \cup \mathscr{E}_s$;
2. $\mathrm{init} = (q_\alpha, true)$, *where* $S = \underline{\mathrm{init}} : \alpha.S + S'$;
3. $\mathscr{TC} = \{\!\!\{ (q_\alpha, \mathbf{1}_{iv(i_S)}, r \cdot [\![I]\!]) \mid \alpha = (i_S, r, I) \in \mathsf{is}(S) \}\!\!\}$,
4. $\mathscr{TD} = \{(q_{\alpha_1}, q_{\alpha_2}, 1, true, true, \underline{\mathrm{a}}) \mid \underline{\mathrm{a}} \in \mathsf{ev}(S) \cap \mathscr{E}_d \wedge \alpha_1 \in \mathsf{is}(S) \wedge S = \underline{\mathrm{a}} : \alpha_2.S + S'\}$
5. $\mathscr{TS} = \{\!\!\{ (q_{\alpha_1}, q_{\alpha_2}, true, true, act(\bar{\mathrm{a}}), \bar{\mathrm{a}}) \mid \bar{\mathrm{a}} \in \mathsf{ev}(S) \cap \mathscr{E}_s \wedge \alpha_1 \in \mathsf{is}(S) \wedge S = \underline{\mathrm{a}} : \alpha_2.S + S' \}\!\!\}$

Once a TDSHA is generated for each subcomponents, the TDSHA of the full uncontrolled system can be built by applying the product construction of TDSHA.

Definition 19. *If* $\Sigma \stackrel{\mathrm{def}}{=} S_1(\mathscr{V}) \bowtie_* \ldots \bowtie_* S_s(\mathscr{V})$ *then* $\mathscr{T}(\Sigma) = \mathscr{T}(S_1(\mathscr{V})) \otimes_* \ldots \otimes_* \mathscr{T}(S_s(\mathscr{V}))$.

Dealing with the controller is simpler, as controllers are finite state automata which impose causality on the happening of events. Event conditions are assigned to edges of TDSHA associated with controllers. All events will be properly dealt with through this construction, as they all appear in the controller since the stochastic HYPE model is well-defined.

First, consider a sequential controller $M = \sum_i \mathrm{a}_i.M_i$. The derivative set of M is defined recursively by $ds(M) = \{M\} \cup \bigcup_i ds(M_i)$.

Definition 20. *Let* $(ConSys, \mathscr{V}, IN, IT, \mathscr{E}_d, \mathscr{E}_s, \mathscr{A}, ec, iv, EC, ID)$ *be a stochastic HYPE model with sequential controller* M. *Then* $\mathscr{T}(M) = (Q, \mathbf{X}, \mathscr{TC}, \mathscr{TD}, \mathscr{TS}, \mathrm{init}, \mathscr{E})$, *the TDSHA associated with* M, *is defined by*

1. $Q = \{q_{M'} \mid M' \in ds(M)\}$; $\mathbf{X} = \mathscr{V}$; $\mathscr{E} = \mathscr{E}_d \cup \mathscr{E}_s$;
2. $\mathrm{init} = (q_M, R_{\underline{\mathrm{init}}})$, *where* $R_{\underline{\mathrm{init}}}$ *is the reset associated with the* $\underline{\mathrm{init}}$ *event.*
3. $\mathscr{TC} = \emptyset$;
4. $\mathscr{TD} = \{(q_{M_1}, q_{M_2}, 1, act(\underline{\mathrm{a}}), R_{\underline{\mathrm{a}}}, \underline{\mathrm{a}}) \mid M_1 = \underline{\mathrm{a}}.M_2,\ M_1, M_2 \in ds(M),\ \underline{\mathrm{a}} \in \mathscr{E}_d,\ ec(\underline{\mathrm{a}}) = (act(\underline{\mathrm{a}}), res(\underline{\mathrm{a}}))\}$;
5. $\mathscr{TS} = \{\!\!\{ (q_{M_1}, q_{M_2}, true, R_{\bar{\mathrm{a}}}, act(\bar{\mathrm{a}}), \bar{\mathrm{a}}) \mid M_1 = \bar{\mathrm{a}}.M_2,\ M_1, M_2 \in ds(M),\ \bar{\mathrm{a}} \in \mathscr{E}_s,\ ec(\bar{\mathrm{a}}) = (act(\bar{\mathrm{a}}), res(\bar{\mathrm{a}})) \}\!\!\}$, *where* $act(\bar{\mathrm{a}}) : \mathbb{R}^{|\mathscr{V}|} \to \mathbb{R}^+$ *is the rate of the transition;*

Definition 21. *Let* $Con = Con_1 \bowtie_L Con_2$ *be a controller. The TDSHA of Con is defined recursively as* $\mathscr{T}(Con) = \mathscr{T}(Con_1) \otimes_L \mathscr{T}(Con_2)$.

The product construction of Definitions 19 and 21 are defined because the factor TDSHA are reset-compatible and init-compatible. This is trivial both for the uncontrolled system (all resets are *true*) and for the controller (resets for the same event are equal). Furthermore, stochastic transitions have consistent rates, as their rate depends only on the labelling event.

Once the TDSHA of the controller and the uncontrolled system are constructed, we simply have to take their product.

Definition 22. *Let* $(ConSys, \mathscr{V}, IN, IT, \mathscr{E}_c, \mathscr{E}_s, \mathscr{A}, ec, iv, EC, ID)$ *be a stochastic HYPE model, with controlled system* $ConSys = \Sigma \bowtie_L \underline{\mathrm{init}}.Con$. *The TDSHA associated with ConSys is*

$$\mathscr{T}(ConSys) = \mathscr{T}(\Sigma) \otimes_L \mathscr{T}(Con).$$

The construction generates TDSHAs with many *unreachable states* [4]. This is a consequence of the fact that sequentiality and causality on actions is imposed just on the final step, when the controller is synchronized with the uncontrolled system. Once the TDSHA is constructed, however, it can be pruned by removing unreachable states. In order to limit combinatorial explosion, one can prune TDSHA's at each intermediate stage. A formal definition of this policy, however, would have made the mapping from stochastic HYPE to TDSHA much more complex.

Theorem 6. *Let* $(ConSys, \mathscr{V}, IN, IT, \mathscr{E}_c, \mathscr{E}_s, \mathscr{A}, ec, iv, EC, ID)$ *be a stochastic HYPE model then* $\mathscr{T}(ConSys) = \mathscr{T}_{SOS}(ConSys)$ *when only reachable states are considered.*

Proof sketch. In order to prove the theorem, we will exhibit a graph isomophism between the discrete graph of the two TDSHA $\mathscr{T}(ConSys)$ and $\mathscr{T}_{SOS}(ConSys)$. The additional information labelling edges of the graph (i.e. guards, resets, rates and priorities) are automatically the same due to the fact they are defined globally. Priorities, in particular, are always equal to 1 (and the min in the product of TDSHA preserves this).

First of all, consider $\mathscr{T}(ConSys)$ and observe that by Definitions 22, 23, and 26, each mode of the automaton contains the set of active influences, which are in bijection with the set of continuous transitions of $\mathscr{T}(ConSys)$. Furthermore, by Definitions 24, 25, and 26, it also contains the current state of each sequential component of the controller. Hence, we can indicate a mode q by the tuple $((\iota_1, r_{1,j_1}, I_{1,j_1}), \ldots, (\iota_k, r_{k,j_k}, I_{k,j_k}), Con_{1,i_1}, \ldots, Con_{h,i_h})$. On the other hand, a mode of $\mathscr{T}_{SOS}(ConSys)$ is of the form $\langle \Sigma \underset{*}{\bowtie} Con, \sigma \rangle$, with $Con = Con_{1,i_1} \underset{*}{\bowtie} \ldots \underset{*}{\bowtie} Con_{h,i_h}$ and where the state is $\sigma = \{(\iota_1, r_{1,j_1}, I_{1,j_1}), \ldots, (\iota_k, r_{k,j_k}, I_{k,j_k})\}$. Therefore, we can define the function ρ mapping each $\langle \Sigma \underset{*}{\bowtie} Con, \sigma \rangle$ to the tuple $((\iota_1, r_{1,j_1}, I_{1,j_1}), \ldots, (\iota_k, r_{k,j_k}, I_{k,j_k}), Con_{1,i_1}, \ldots, Con_{h,i_h})$. This function is well-defined, as the derivative of the controller plus the state uniquely identify the configuration, and it is easily seen to be a bijection.

In order to show that ρ is a graph isomorphism, we need to show that if $\langle P_1, \sigma_1 \rangle \xrightarrow{\mathrm{a}} \langle P_2, \sigma_2 \rangle$ is a transition of the LTS, then there is a discrete or stochastic transition (depending on a) in $\mathscr{T}(ConSys)$ of the form $(\rho(\langle P_1, \sigma_1 \rangle), \rho(\langle P_2, \sigma_2 \rangle), \cdot, \cdot, \cdot, \mathrm{a})$, with matching multiplicity.

This can be easily seen by structural induction on agents, focussing on synchronisation and taking subcomponents or sequential controllers as base cases. Inspecting Definitions 22 and 24, it is easy to see that the previous property holds for the base cases, as those definitions construct the TDSHA by considering all transitions and all states of the LTS.

As for synchronization, we just need to notice that the product construction for TDSHA perfectly matches the SOS rules in terms of updated components and updated states, hence a simple structural induction argument will do. Multiplicity of stochastic transitions is also preserved as the product construction acts on multi-graphs for what concerns stochastic events. Hence, we have proved that ρ is a graph isomorphism. The fact that reachable states are the same in both graphs then follows by a simple induction on the distance in such graphs from the initial mode. □

D Bisimulation for TDSHA

We have a number of choices when we consider bisimulations over TDSHA in light of definitions for PDMP [32]. We can consider a single discrete jump as a transition and match or ignore labels, or we can consider a sequence of discrete jumps, ignoring labels or considering each jump to be labelled with $\underline{\tau}$ except for one labelled with $\underline{\mathrm{a}} \neq \underline{\tau}$. The first two options can be viewed as strong forms of bisimulation and the last two as weak forms of bisimulation. One reason for ignoring labels is then we can compare any equivalence we define with those for PDMPs which have no labels. However, the focus of this paper is stochastic HYPE where labels are used, and hence we focus on those with labels.

We start by giving some definitions that we require for strong bisimulation. We need a notion of measurable relation which will allow the definition of function match_B that provides a bijection between two quotient spaces and can be used in the definition of bisimulation.

D.1 Measurable relations

In order to define bisimulations for TDSHA, we need to introduce a notion of measurable relation, taken from [32]. In the following, we let X,Y be separable metric spaces and $\mathscr{X},\mathscr{Y}$ be the corresponding Borel sigma-algebras, so that $(X,\mathscr{X})$ and $(Y,\mathscr{Y})$ are two Borel measurable spaces.

We consider a relation $B \subseteq X \times Y$, and we assume that the projections on the two components coincide with the whole spaces, i.e. $\{x \in X \mid \exists y \in Y, (x,y) \in B\} = X$ and $\{y \in Y \mid \exists x \in X, (x,y) \in B\} = Y$. Then we define two equivalence relations, one on X and one on Y. The equivalence relation B_X on X is defined as the transitive closure of the relation $\{(x_1,x_2) \mid \exists y \in Y, (x_1,y),(x_2,y) \in B\}$. Similarly for B_Y.

A straightforward property of the equivalence relations induced by B would be so that the two quotient spaces $X/B = X/B_X$ and $Y/B = Y/B_Y$ are in bijection. In fact, the map $\mathsf{match}_B : X/B \to Y/B$ such that $\mathsf{match}_B([x]) = [y]$ if and only if $(x,y) \in B$, is a well-defined bijection.

In the following, we denote with π_X the canonical projection of X onto X/B, defined by $\pi_X(x) = [x]$. Similarly for Y.

Another property of the equivalence relations induced by B is that X/B and Y/B inherit the sigma-algebra structure from X and Y. In fact, it is straightforward to check that the collection $\mathscr{X}/B$ of subsets of X/B, containing the sets $\{[x] \in A \mid A \in \mathscr{X}\}$, is a sigma-algebra.

Definition 23. *The relation $B \subseteq X \times Y$ is* measurable *if and only if, for each $A \in \mathscr{X}/B$, it holds that $\mathsf{match}_B(A) \in \mathscr{Y}/B$, and vice versa.*

We further need the notion of equivalent probability measures on X and Y, with respect to a measurable relation B. Essentially, two probability measures will be equivalent if and only if they will induce the same probability distribution on the quotient sets X/B and Y/B.

Definition 24. *Let P_X be a probability measure on $(X,\mathscr{X})$ and P_Y be a probability measure on $(Y,\mathscr{Y})$. P_X and P_Y are equivalent with respect to the measurable relation B if and only if, for each $A \in \mathscr{X}/B$, it holds that*

$$P_X(\pi_X^{-1}(A)) = P_Y(\pi_Y^{-1}(\mathsf{match}_B(A))).$$

D.2 Bisimulation

To define TDSHA bisimulation, definitions are required to probailities of actions. The two definitions below define transitions that involve actions from a set A, and calculate probabilities for these transitions. The set A is used to determine which labels are matched and for matching of single labels, $\{a\}$ can be used. We can also define non-singleton subsets of A but we defer these to further work.

Definition 25. *Given a TDSHA $\mathscr{T} = (Q,\mathbf{X},\mathscr{TC},\mathscr{TD},\mathscr{TS},\mathrm{init},\mathscr{E})$, let*

$$\mathscr{TS}((q,\mathbf{x}),A) = \{\!\{\, \eta \in \mathscr{TS} \mid q_1^\eta = q, e_\eta \in A, g_\eta(\mathbf{x}) = true \,\}\!\}$$

be the set of stochastic transitions with labels in $A \subseteq \mathscr{E}$ active in $(q,\mathbf{x})$.
Furthermore, let

$$\mathscr{TD}((q,\mathbf{x}),A) = \{\!\{\, \delta \in \mathscr{TD} \mid q_1^\delta = q, e_\delta \in A, g_\delta(\mathbf{x}) = true \,\}\!\}$$

be the set of instantaneous transitions with labels in $A \subseteq \mathscr{E}$ active in $(q,\mathbf{x})$.

Let $\lambda(q,\mathbf{x}) = \sum\{\!\{ f_\eta(\mathbf{x}) \mid \eta \in \mathscr{TS}((q,\mathbf{x}),\mathscr{E}) \}\!\}$ and let $w(q,\mathbf{x}) = \sum\{\!\{ w_\delta(\mathbf{x}) \mid \delta \in \mathscr{TD}((q,\mathbf{x}),\mathscr{E}) \}\!\}$.

Definition 26. *The (1-step)* probability of a stochastic transition with a label in A from $(q,\mathbf{x})$ to a set C *is defined as*

$$P_{1s}^{\mathscr{TS}}((q,\mathbf{x}),A,C) = \sum\{\!\{ Pr\{(q_2^\eta, r_\eta(\mathbf{x},\mathbf{W})) \in C\} \cdot f_\eta(\mathbf{x})/\lambda(q,\mathbf{x}) \mid \eta \in \mathscr{TS}((q,\mathbf{x}),A) \}\!\}$$

for $\lambda(q,\mathbf{x}) \neq 0$ *and 0 otherwise.*

Similarly, the (1-step) probability of an instantaneous transition a label in A from $(q,\mathbf{x})$ to a set C *is defined as*

$$P_{1s}^{\mathscr{TD}}((q,\mathbf{x}),A,C) = \sum \{\!| Pr\{(q_2^{\delta}, r_{\eta}(\mathbf{x},\mathbf{W})) \in C\} \cdot w_{\delta}/\omega(q,\mathbf{x}) \mid \delta \in \mathscr{TD}((q,\mathbf{x}),A) |\!\}$$

for $\omega(q,\mathbf{x}) \neq 0$*, and 0 otherwise.*

To improve the readability of the following definitions, we define the predicate

$$G(q,\mathbf{x}) = \bigvee \{g_{\delta}(\mathbf{x}) \mid \delta \in \mathscr{TD}, q_1^{\delta} = q\},$$

which is true when at least one guard of an instantaneous transition is true. Let $\phi(t,\mathbf{X})$ denote the solution of the ODEs of the TDHSA taking into account the initial values of variables.

Next, we define a bisimulation that matches on individual labels.

Definition 27. *Let* $\mathscr{T}_i = (Q_i, \mathbf{X_i}, \mathscr{TC}_i, \mathscr{TD}_i, \mathscr{TS}_i, \text{init}_i, \mathscr{E}_i)$*,* $i = 1,2$ *be two TDSHAs, and let the relation* $B \subseteq (Q_1 \times \mathbb{R}^{n_1}) \times (Q_2 \times \mathbb{R}^{n_2})$ *be a measurable relation. B is a* labelled TDHSA bisimulation *for* $\mathscr{T}_1$ *and* $\mathscr{T}_2$ *whenever for all* $((q_1,\mathbf{x}),(q_2,\mathbf{x})) \in B$*,*

1. *Assuming two smooth output functions* $\mathsf{out}_i : \mathbb{R}^{n_i} \to \mathbb{R}^m$*, for* $i = 1,2$ *with* $m \leq n_i$ *then* $\mathsf{out}_1(\mathbf{x_1}) = \mathsf{out}_2(\mathbf{x_2})$
2. $\lambda(q_1,\mathbf{x_1}) = \lambda(q_2,\mathbf{x_2})$
3. $G_1(q_1,\mathbf{x_1}) = G_2(q_2,\mathbf{x_2})$
4. *For all* $0 \leq t \leq t^*$*,* $((q_1,\phi(t,\mathbf{x_1})),(q_2,\phi(t,\mathbf{x_2})) \in B$ *where* t^* *is the smallest value such that* $G_1(q_1,\phi(t,\mathbf{x_1})) = G_2(q_2\phi(t,\mathbf{x_2})) = true$
5. *For all* $C \in (Q_1 \times \mathbb{R}^{n_1})/B$*, and for all* $a \in \mathscr{E}_1 \cup \mathscr{E}_2$*,*
   $P_{1s}^{\mathscr{TS}}((q_1,\mathbf{x_1}),\{a\},C) = P_{1s}^{\mathscr{TS}}((q_2,\mathbf{x_2}),\{a\},\mathsf{match}_B(C))$ *and*
   $P_{1s}^{\mathscr{TD}}((q_1,\mathbf{x_1}),\{a\},C) = P_{1s}^{\mathscr{TD}}((q_2,\mathbf{x_2}),\{a\},\mathsf{match}_B(C))$.

Definition 28. $\mathscr{T}_1$ *and* $\mathscr{T}_2$ *are* labelled TDSHA bisimilar*, written* $\mathscr{T}_1 \sim_T^l \mathscr{T}_2$*, whenever there exists B a TDSHA bisimulation for* $\mathscr{T}_1$ *and* $\mathscr{T}_2$*.*

We next consider the relationship between bisimulation over stochastic HYPE models and TDSHAs.

Theorem 7. *Let* $(P_i, \mathscr{V}, IN, IT, \mathscr{E}_d, \mathscr{E}_s, \mathscr{A}, ec, iv, EC, ID)$ *for* $i = 1,2$*, be two stochastic HYPE models whose TDHSAs are* $\mathscr{T}_i$*, if* $P_1 \sim_{\mathbf{sm}}^{\doteq} P_2$ *then* $\mathscr{T}_1 \sim_T^l \mathscr{T}_2$*.*

Proof sketch. The stochastic HYPE models P_i can be transformed as described in Definition 7 to TDSHA $\mathscr{T}_i = (\text{ds}(P_i), \mathscr{V}, \mathscr{TC}_i, \mathscr{TD}_i, \mathscr{TS}_i, (\langle R_i, \sigma_i\rangle, v_i), \mathscr{E}_d \cup \mathscr{E}_s)$. Let the equivalence relation B be defined by

$$B = \left\{ \left((\langle Q_1, \sigma_1\rangle, \mathbf{x}), (\langle Q_2, \sigma_2\rangle, \mathbf{x}) \right) \mid Q_1 \sim_{\mathbf{sm}}^{\doteq} Q_2, \sigma_1 \doteq \sigma_2 \right\}.$$

To satisfy the first condition, let out_i be the identity function.

Next, note that $\lambda(\langle Q_i, \sigma_i\rangle, \mathbf{x}) = \sum\{r(\langle Q_i, \sigma_i\rangle, \bar{\mathrm{a}}, C) | \bar{\mathrm{a}} \in \mathscr{E}_s, C \in \mathscr{F}/(\sim_{\mathbf{sm}}^{\doteq})\}$. Since $Q_1 \sim_{\mathbf{sm}}^{\doteq} Q_2$ and $\sigma_1 \doteq \sigma_2$, $r(\langle Q_1, \sigma_1\rangle, \bar{\mathrm{a}}, C) = r(\langle Q_2, \sigma_2\rangle, \bar{\mathrm{a}}, C)$. Hence it is straightforward to show that $\lambda(\langle Q_1, \sigma_1\rangle, \mathbf{x_1}) = \lambda(\langle Q_2, \sigma_2\rangle, \mathbf{x_2})$.

For $Q_1 \sim^{\doteq}_{\mathbf{sm}} Q_2$ and $\sigma_1 \doteq \sigma_2$, any transition that can be performed by $\langle Q_1, \sigma_1 \rangle$ can be matched by $\langle Q_2, \sigma_2 \rangle$ hence the same events must be available in both configurations. $G(\langle Q_1, \sigma_1 \rangle, \mathbf{x}) = \bigvee\{act(\underline{a})(\mathbf{x}) \mid \langle Q_1, \sigma_1 \rangle \xrightarrow{\underline{a}}\} = \bigvee\{act(\underline{a})(\mathbf{x}) \mid \langle Q_2, \sigma_2 \rangle \xrightarrow{\underline{a}}\} = G(\langle Q_2, \sigma_2 \rangle, \mathbf{x})$.

By Theorem 3, $(Q_1)_{\sigma_1} = (Q_2)_{\sigma_2}$, for $Q_1 \sim^{\doteq}_{\mathbf{sm}} Q_2$ and $\sigma_1 \doteq \sigma_2$ hence we can conclude that the pair $((\langle Q_1, \sigma_1 \rangle, \phi(t, \mathbf{x})), (\langle Q_2, \sigma_2 \rangle, \phi(t, \mathbf{x}))) \in B$ where $\phi(t, \mathbf{x})$ is the solution of the ODEs given by $(Q_1)_{\sigma_1}$ and for all $0 \leq t \leq t^*$ where t^* is the smallest value that $G(\langle Q_1, \sigma_1 \rangle, \mathbf{x}) = G(\langle Q_2, \sigma_2 \rangle, \mathbf{x}) = \mathit{true}$.

Finally, we need to show that for all $C \in (\mathrm{ds}(P_1) \times \mathbb{R}^n)/B$, and for all $a \in \mathscr{E}$,

- $P_{1s}^{\mathscr{TS}}((\langle Q_1, \sigma_1 \rangle, \mathbf{x}), \{\bar{a}\}, C) = P_{1s}^{\mathscr{TS}}((\langle Q_2, \sigma_2 \rangle, \mathbf{x}), \{\bar{a}\}, \mathsf{match}_B(C))$
- $P_{1s}^{\mathscr{TD}}((\langle Q_1, \sigma_1 \rangle, \mathbf{x}), \{\underline{a}\}, C) = P_{1s}^{\mathscr{TD}}((\langle Q_2, \sigma_2 \rangle, \mathbf{x}), \{\underline{a}\}, \mathsf{match}_B(C))$

Since B is the identity relation over $\mathbb{R}^n$, each equivalence class C is of the form $[\langle Q_1', \sigma_1' \rangle] \times \{\mathbf{x}\}$ and $\mathsf{match}_B(C)$ is of the form $[\langle Q_2', \sigma_2' \rangle] \times \{\mathbf{x}\}$, where $Q_1' \sim^{\doteq}_{\mathbf{sm}} Q_2'$. The first item follows from the fact that $r(\langle Q_1, \sigma_1 \rangle, \bar{a}, [\langle Q_1', \sigma_1' \rangle]) = r(\langle Q_2, \sigma_2 \rangle, \bar{a}, [\langle Q_2', \sigma_2' \rangle])$ and reset functions depend only on $\bar{a}$. The second is a consequence of the fact that $P_{1s}^{\mathscr{TD}}((\langle Q_i, \sigma_i \rangle, \mathbf{x}), \{\underline{a}\}, C)$ is proportional to the number of distinct derivatives in C that $\langle Q_i, \sigma_i \rangle$ has after an $\underline{a}$ event. Since each $\langle Q_i, \sigma_i \rangle$ must be able to match the transitions of the other, they have the same number of distinct derivatives. Hence we can conclude that B is a labelled TDSHA bisimulation. □

The converse of this theorem does not hold since $\doteq$ is defined for a specific variable, whereas the bisimulation that is constructed for the TDSHA can sum across multiple variables. Consider the following simple counterexample with two one-mode stochastic HYPE systems, with two continuous variables X and Y, two influences (ι_X acting on X and ι_Y acting on Y), and no events.

$$\begin{array}{llll} A_1 \stackrel{def}{=} \underline{\mathrm{init}} : (\iota_X, a, \mathit{const}) & & B_1 \stackrel{def}{=} \underline{\mathrm{init}} : (\iota_Y, b, \mathit{const}) \\ A_2 \stackrel{def}{=} \underline{\mathrm{init}} : (\iota_X, a+b, \mathit{const}) & & B_2 \stackrel{def}{=} \underline{\mathrm{init}} : (\iota_Y, 0, \mathit{const}) \end{array}$$

The respective controlled systems are $P_i = A_i \bowtie_* B_i$ corresponding to the ODE systems $\frac{d}{dt}(X,Y) = (a,b)$ and $\frac{d}{dt}(X,Y) = (a+b,0)$. Looking at the TDSHA, it is easy to see that $\mathscr{T}_{P_1} \sim^{\ell}_{T} \mathscr{T}_{P_2}$, as the equivalence relation $B = \{((x,y),(x+y,0))\}$ is a TDSHA bisimulation (using $\mathsf{out}_1(X,Y) = X+Y$ and $\mathsf{out}_2(X,Y) = X$ as output functions, conditions 1 and 4 follow easily, while the others are trivially true as there is no discrete jump). However, it does not hold that $P_1 \sim^{\doteq}_{\mathbf{sm}} P_2$, as $\doteq$ requires the ODEs in each matching mode to be the same.

D.3 TDHSA bisimulation applied to the example

We know consider how TDHSA bisimulation can be used for the example. There are different ways to implement the timing of the system. We can remove the Timer_i subcomponents from within Sys and add the following new timer component $\mathit{Timer} \stackrel{def}{=} \underline{\mathrm{init}} : (t, 1, \mathit{const}).\mathit{Timer}$ with $iv(t) = T$, T a new variable (and without influences t_i but keeping variables T_i). Various event conditions must be modified as follows

$$\begin{array}{rcl} ec(\underline{\mathrm{init}}) & = & (\mathit{true}, P' \sim P_0 \wedge T_1' \sim 0 \wedge T_2' \sim 0 \wedge T' \sim 0 \wedge B' \sim B_0 \wedge M' \sim 0) \\ ec(\underline{\mathrm{assem}_i}) & = & (T \geq T_i' + \mathit{atime}_i, B' \sim B + m_i) \end{array}$$

Denote this new system by $\mathit{Assembler}_T = \mathit{Sys}_T \bowtie_* \underline{\mathrm{init}}.\mathit{Con}$ where

$$\begin{array}{rcl} \mathit{Sys}_T & \stackrel{def}{=} & (\mathit{Feed}_1 \bowtie_* \mathit{Feed}_2 \bowtie_* \mathit{Feed}_3) \bowtie_* \mathit{Inspect} \bowtie_* \\ & & \mathit{Machine}_1(W_1) \bowtie_* \mathit{Machine}_2(W_2) \bowtie_* \mathit{Timer}. \end{array}$$

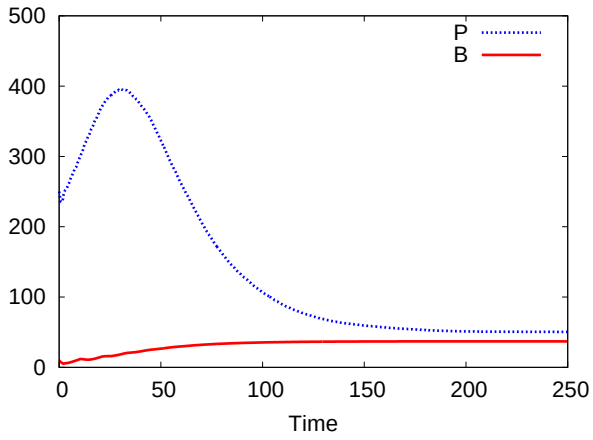

Figure 11: Average values for $Assembler_T$ over 10000 simulations using the same parameters as in Figure 7.

We can show that $\mathscr{T}(Assembler_T) \sim_T^l \mathscr{T}(Assembler)$. In order to do this, first observe that the LTSs of the two stochastic HYPE models are isomorphic. This holds because in each configuration $\langle P, \sigma \rangle$ the state σ is determined by the local state of the controlled system P, specifically by the local state of the controller in P. Hence, the map $\rho : \langle Sys \bowtie_* Con, \sigma_1 \rangle \mapsto \langle Sys_T \bowtie_* Con, \sigma_T \rangle$, where σ_1 and σ_T depend on Con, is a bijection. Given a configuration $\langle P, \sigma \rangle$, we write $AProc_i$ in P if and only if $P = Sys_1 \bowtie_* (AProc_i \bowtie_* Con')$.

When we map these stochastic HYPE models to the corresponding TDSHA, we therefore obtain two automata with the same discrete skeleton and the same event set, but with different variables, different ODEs within modes, and different guards and resets for some events. To prove that they are bisimilar, we need to exhibit a measurable relation B satisfying the conditions of Definition 27. Let $(\langle P, \sigma \rangle, \mathbf{x}, s_1, s_2) \in Q \times \mathbb{R}^6$ be a state of TDSHA $\mathscr{T}(Assembler)$, where s_i is the value of timer T_i, and $(\rho(\langle P, \sigma \rangle), \mathbf{x}, t, t_1, t_2) \in Q \times \mathbb{R}^7$ be a state of TDSHA $\mathscr{T}(Assembler_T)$, where t is the value of T, and t_i is the value of variable T_i. Now, if $AProc_i$ in P (and only in this case), then it is easy to check that the value of T_i in $\mathscr{T}(Assembler)$ has to be equal to $T - T_i$ in $\mathscr{T}(Assembler_T)$, as both expressions measure the time elapsed since the firing of event $\underline{\text{remove}}_i$. This suggests the following relation which is easily seen to be measurable.

$$B = \{((\langle P, \sigma \rangle, \mathbf{x}, s_1, s_2), (\rho(\langle P, \sigma \rangle), \mathbf{x}, t, t_1, t_2)) \mid AProc_i \text{ in } P \Rightarrow t - t_i = s_i\},$$

In order for B to be a TDSHA bisimulation, we need to ignore timer values while comparing states. This is obtained by taking the out functions to be the projections over the remaining variables: $\mathtt{out}_1(\mathbf{x}, s_1, s_2) = \mathbf{x}$ and $\mathtt{out}_T(\mathbf{x}, t, t_1, t_2) = \mathbf{x}$. Considering Definition 27, condition 1 follows from the fact that the vector fields, restricted to non-timer variables, coincide. Condition 2 is trivial, as only instantaneous events have been modified, while the condition 3 on guards is a consequence of the definition of B in the states where timer i is active. In particular, the activation time t^* coincides in both models, and so condition 4 follows. Finally, condition 5 stems from the isomorphism of LTS and the fact that the variables of T_i are both reset after event $\underline{\text{remove}}_i$.

By contrast, the two HYPE models *Assembler* and $Assembler_T$ are trivially not system bisimilar. This is easily seen by inspecting Definition 16, which requires variable sets and event condition to be the same in both models.