Extending Hybrid CSP with Probability and Stochasticity

Yu Peng, Shuling Wang$^{(\boxtimes)}$, Naijun Zhan, and Lijun Zhang

State Key Laboratory of Computer Science, Institute of Software,
Chinese Academy of Sciences, Beijing, China
wangsl@ios.ac.cn

Abstract. Probabilistic and stochastic behavior are omnipresent in computer controlled systems, in particular, so-called safety-critical hybrid systems, because of fundamental properties of nature, uncertain environments, or simplifications to overcome complexity. Tightly intertwining discrete, continuous and stochastic dynamics complicates modelling, analysis and verification of stochastic hybrid systems (SHSs). In the literature, this issue has been extensively investigated, but unfortunately it still remains challenging as no promising general solutions are available yet. In this paper, we give our effort by proposing a general compositional approach for modelling and verification of SHSs. First, we extend Hybrid CSP (HCSP), a very expressive and process algebra-like formal modeling language for hybrid systems, by introducing probability and stochasticity to model SHSs, which is called stochastic HCSP (SHCSP). To this end, ordinary differential equations (ODEs) are generalized by stochastic differential equations (SDEs) and non-deterministic choice is replaced by probabilistic choice. Then, we extend Hybrid Hoare Logic (HHL) to specify and reason about SHCSP processes. We demonstrate our approach by an example from real-world.

1 Introduction

Probabilistic and stochastic behavior are omnipresent in computer controlled systems, such as safety-critical hybrid systems, because of uncertain environments, or simplifications to overcome complexity. For example, the movement of aircrafts could be influenced by wind; in networked control systems, message loss and other random effects (e.g., node placement, node failure, battery drain, measurement imprecision) may happen.

Stochastic hybrid systems (SHSs) are systems in which discrete, continuous and stochastic dynamics tightly intertwine. As many of SHSs are safety-critical, a thorough validation and verification activity is necessary to enhance the quality of SHSs and, in particular, to fulfill the quality criteria mandated by the relevant standards. But modeling, analysis and verification of SHSs is difficult and challenging. An obvious research line is to extend hybrid automata [10], which is the most popular model for traditional hybrid systems, by adding probability and

X. Li et al. (Eds.): SETTA 2015, LNCS 9409, pp. 87–102, 2015.
DOI: 10.1007/978-3-319-25942-0_6

stochasticity. Then, verification of SHSs can be done naturally through reachability analysis, either by probabilistic model-checking [1–3,6,8,20,21], or by simulation i.e., statistical model-checking [15,23]. Along this line, several different notions of *stochastic hybrid automata* have been proposed [1–3,6,8,20,21], with the difference on where to introduce randomness. One option is to replace deterministic jumps by probability distribution over deterministic jumps. Another option is to generalize differential equations inside a mode by stochastic differential equations. Stochastic hybrid systems comprising stochastic differential equations have been investigated in [1,5,13]. More general models can be obtained by mixing the above two choices, and by combining them with memoryless timed probabilistic jumps [4], with a random reset function for each discrete jump [6]. An overview of this line can be found in [4].

To model complex systems, some compositional modelling formalisms have been proposed, e.g., HMODEST [7] and stochastic hybrid programs [18]. HCSP due to He, Zhou, et al [9,22] is an extension of CSP [12] by introducing differential equations to model continuous evolution and three types of interruptions (i.e., communication interruption, timeout and boundary condition) to model interactions between continuous evolutions and discrete jumps in HSs. The extension of CSP to probabilistic setting has been investigated by Morgan et al. [16]. In this paper, we propose a compositional approach for modelling and verification of stochastic hybrid systems. First, we extend Hybrid CSP (HCSP), a very expressive and process algebra-like modeling language for hybrid systems by introducing probability and stochasticity, called stochastic HCSP (SHCSP), to model SHSs. In SHCSP, ordinary differential equations (ODEs) are generalized to stochastic differential equations (SDEs), and non-deterministic choice is replaced by probabilistic choice. Different from Platzer's work [18], SHCSP provides more expressive constructs for describing hybrid systems, including communication, parallelism, interruption, and so on.

Probabilistic model-checking of SHSs does not scale, in particular, taking SDEs into account. For example, it is not clear how to approximate the reachable sets of a simple linear SDEs with more than two variables. Therefore, existing verification techniques based on reachability analysis for SHSs are inadequate, and new approaches are expected. As an alternative, in [18], Platzer for the first time investigated how to extend deductive verification to SHSs. Inspired by Platzer's work, for specifying and reasoning about SHCSP process, we extend Hybrid Hoare Logic [14], which is an extension of Hoare logic [11] to HSs, to SHSs. Comparing with Platzer's work, more computation features of SHSs, and more expressive constructs such as concurrency, communication and interruption, can be well handled in our setting. We demonstrate our approach by modeling and verification of the example of aircraft planning problem from the real-world.

2 Background and Notations

Assume that $\mathcal{F}$ is a σ-algebra on set Ω and P is a probability measure on $(\Omega, \mathcal{F})$, then $(\Omega, \mathcal{F}, P)$ is called a *probability space*. We here assume that every subset of a null set (i.e., $P(A) = 0$) with probability 0 is measurable. A property which holds with probability 1 is said to hold *almost surely* (*a.s.*). A *filtration* is a sequence of σ-algebras $\{\mathcal{F}_t\}_{t\geq 0}$ with $\mathcal{F}_{t_1} \subseteq \mathcal{F}_{t_2}$ for all $t_1 < t_2$. We always assume that a filtration $\{\mathcal{F}_t\}_{t\geq 0}$ has been completed to include all null sets and is right-continuous.

Let $\mathcal{B}$ represent the Borel σ-algebra on $\mathbb{R}^n$, i.e. the σ-algebra generated by all open subsets. A mapping $X : \Omega \to \mathbb{R}^n$ is called $\mathbb{R}^n$-valued *random variable* if for each $B \in \mathcal{B}$, we have $X^{-1}(B) \in \mathcal{F}$, i.e. X is *$\mathcal{F}$-measurable.* A *stochastic process* X is a function $X : T \times \Omega \to \mathbb{R}^n$ such that for each $t \in T$, $X(t,\cdot) : \Omega \to \mathbb{R}^n$ is a random variable, and for each $\omega \in \Omega$, $X(\cdot,\omega) : T \to \mathbb{R}^n$ corresponds to a *sample path.* A stochastic process X is *adapted* to a filtration $\{\mathcal{F}_t\}_{t\geq 0}$ if X_t is $\mathcal{F}_t$-measurable. Intuitively, a filtration represents all available historical information of a stochastic process, but nothing related to its future. A *càdlàg* function defined on $\mathbb{R}$ is *right continuous* and has *left limit.* A stochastic process X is *càdlàg* iff all of its paths $t \to X_t(\omega)$ (for each $\omega \in \Omega$) are *càdlàg.* A *d*-dimensional *Brownian motion* W is a stochastic process with $W_0 = 0$ that is continuous almost surely everywhere and has independent increments with time, i.e. $W_t - W_s \sim N(0, t-s)$ (for $0 \leq s < t$), where $N(0, t-s)$ denotes the normal distribution with mean 0 and variance $t - s$. Brownian motion is mathematically extremely complex. Its path is almost surely continuous everywhere but differentiable nowhere. Intuitively, W can be understood as the limit of a random walk. A *Markov time* with respect to a stochastic process X is a random variable τ such that for any $t \geq 0$, the event $\{\tau \leq t\}$ is determined by (at most) the information up to time t, i.e. $\{\tau \leq t\} \in \mathcal{F}_t$.

We use *stochastic differential equation* (SDE) to model stochastic continuous evolution, which is of the form $dX_t = b(X_t)dt + \sigma(X_t)dW_t$, where W_t is a Brownian motion. In which, the drift coefficient $b(X_t)$ determines how the deterministic part of X_t changes with respect to time and the diffusion coefficient $\sigma(X_t)$ determines the stochastic influence to X_t with respect to the Brownian motion W_t. Obviously, any solution to an SDE is a stochastic process.

3 Stochastic HCSP

A system in Stochastic HCSP (SHCSP) consists of a finite set of sequential processes in parallel which communicate via channels synchronously. Each sequential process is represented as a collection of stochastic processes, each of which arises from the interaction of discrete computation and stochastic continuous dynamics modeled by stochastic differential equations.

Let *Proc* represent the set of SHCSP processes, Σ the set of channel names. The syntax of SHCSP is given as follows:

$$\begin{aligned} P ::=\ & \mathbf{skip} \mid x := e \mid ch?x \mid ch!e \mid P; Q \mid B \to P \mid P^* \\ & \mid P \sqcup_p Q \mid \langle ds = bdt + \sigma dW \& B\rangle \\ & \mid \langle ds = bdt + \sigma dW \& B\rangle \trianglerighteq [\!]_{i\in I}(\omega_i \cdot ch_i * \to Q_i) \end{aligned}$$

$$S ::= P \mid S \| S$$

Here $ch, ch_i \in \Sigma$, ch_i* stands for a communication event, e.g. $ch?x$ or $ch!e$, x is a variable, B and e are Boolean and arithmetic expressions, $P, Q, Q_i \in Proc$ are sequential processes, $p \in [0,1]$ stands for the probability of the choice between P and Q, s for a vector of continuous variables, b and σ for functions of s, W for the Brownian motion process. At the end, S stands for a system, i.e., a SHCSP process.

As defined in the syntax of P, the processes in the first line are original from HCSP, while the last two lines are new for SHCSP. The individual constructs can be understood intuitively as follows:

- **skip**, the assignment $x := e$, the sequential composition $P; Q$, and the alternative statement $B \rightarrow P$ are defined as usual.
- $ch?x$ receives a value along channel ch and assigns it to x.
- $ch!e$ sends the value of e along channel ch. A communication takes place when both the sending and the receiving parties are ready, and may cause one side to wait.
- The repetition P^* executes P for some finite number of times.
- $P \sqcup_p Q$ denotes probabilistic choice. It behaves as P with probability p and as Q with probability $1 - p$.
- $\langle ds = bdt + \sigma dW \& B\rangle$ specifies that the system evolves according to the stochastic process defined by the stochastic differential equation $ds = bdt + \sigma dW$. As long as the boolean expression B, which defines the *domain of* s, turns false, it terminates. We will later use $d(s)$ to return the dimension of s.
- $\langle ds = bdt + \sigma dW \& B\rangle \trianglerighteq []_{i \in I}(\omega_i \cdot ch_i* \rightarrow Q_i)$ behaves like $\langle ds = bdt + \sigma dW \& B\rangle$, except that the stochastic evolution is preempted as soon as one of the communications ch_i* takes place, after that the respective Q_i is executed. I is supposed to be finite and for each $i \in I$, $\omega_i \in \mathbb{Q}^+$ represents the *weight* of ch_i*. If one or more communications are ready at the same time, say they are $\{ch_j*\}_{j \in J}$ with $J \subseteq I$ and $|J| \geq 1$, then ch_j is chosen with the probability $\frac{\omega_j}{\Sigma_{j \in J}\omega_j}$, for each $j \in J$. If the stochastic dynamics terminates before a communication among $\{ch_i*\}_I$ occurring, then the process terminates without communicating.
- $S_1 \| S_2$ behaves as if S_1 and S_2 run independently except that all communications along the common channels connecting S_1 and S_2 are to be synchronized. The processes S_1 and S_2 in parallel can neither share variables, nor input nor output channels.

3.1 A Running Example

We use SHCSP to model the aircraft position during the flight, which is inspired from [19]. Consider an aircraft that is following a flight path consisting of a sequence of line segments at a fixed altitude. Ideally, the aircraft should fly at a constant velocity v along the nominal path, but due to the wind or cloud disturbance, the deviation of the aircraft from the path may occur. For safety, the aircraft should follow a correction heading to get back to the nominal path as quickly as possible. On one hand, the correction heading should be orthogonal to the nominal path for the shortest way back, but on the other hand, it should also go ahead to meet the destination. Considering these two objectives, we assume the correction heading always an acute angle with the nominal path.

Here we model the behavior of the aircraft along one line segment. Without loss of generality, we assume the segment is along x-axis, with $(x_s, 0)$ as the starting point and $(x_e, 0)$ as the ending point. When the aircraft deviates from the segment with a vertical distance greater than λ, we consider it enters a dangerous state. Let (x_s, y_0) be the initial position of the aircraft in this segment, then the future position of the aircraft $(x(t), y(t))$ is governed by the following SDE:

$$\begin{pmatrix} dx(t) \\ dy(t) \end{pmatrix} = v \begin{pmatrix} cos(\theta(t)) \\ sin(\theta(t)) \end{pmatrix} dt + dW(t)$$

where $\theta(t)$ is the correction heading and is defined with a constant degree $\frac{\pi}{4}$ when the aircraft deviates from the nominal path:

$$\theta(t) = \begin{cases} -\frac{\pi}{4} & \text{if } y(t) > 0 \\ 0 & \text{if } y(t) = 0 \\ \frac{\pi}{4} & \text{if } y(t) < 0 \end{cases}$$

Let B be $x_s \leq x \leq x_e$, the movement of the aircraft described above can be modelled by the following SHCSP process P_{Air}:

$$x = x_s; y = y_0; \langle [dx, dy]^T = v[cos(\theta(t)), sin(\theta(t))]^T dt + dW(t) \& B \rangle$$

4 Operational Semantics

Before giving operational semantics, we introduce some notations first.

System Variables. In order to interpret SHCSP processes, we use non-negative reals $\mathbb{R}^+$ to model time, and introduce a global clock *now* as a system variable to record the time in the execution of a process. A *timed communication* is of the form $\langle ch.c, b\rangle$, where $ch \in \Sigma$, $c \in \mathbb{R}$ and $b \in \mathbb{R}^+$, representing that a communication along channel ch occurs at time b with value c transmitted. The set $\Sigma \times \mathbb{R} \times \mathbb{R}^+$ of all timed communications is denoted by $T\Sigma$. The set of all timed traces is

$$T\Sigma^*_{\leq} = \{\gamma \in T\Sigma^* \mid \text{ if } \langle ch_1.c_1, b_1\rangle \text{ precedes } \langle ch_2.c_2, b_2\rangle \text{ in } \gamma, \text{ then } b_1 \leq b_2\}.$$

If $C \subseteq \Sigma$, $\gamma \upharpoonright_C$ is the projection of γ onto C such that only the timed communications along channels of C in γ are preserved. Given two timed traces γ_1, γ_2, and $X \subseteq \Sigma$, the *alphabetized parallel* of γ_1 and γ_2 over X, denoted by $\gamma_1 \gamma_2$, results in the following set of timed traces

$$\{\gamma \mid \gamma \upharpoonright_{\Sigma-(\Sigma(\gamma_1)\cup\Sigma(\gamma_2))} = \epsilon, \gamma \upharpoonright_{\Sigma(\gamma_1)} = \gamma_1, \gamma \upharpoonright_{\Sigma(\gamma_2)} = \gamma_2 \text{ and } \gamma \upharpoonright_X = \gamma_1 \upharpoonright_X = \gamma_2 \upharpoonright_X\},$$

where $\Sigma(\gamma)$ stands for the set of channels that occur in γ.

To model synchronization of communication events, we need to describe their readiness. Because a communication itself takes no time when both parties get ready, thus, at a time point, multiple communications may occur. In order to record the execution order of communications occurring at the same time point, we prefix each communication readiness a timed trace that happened before the ready communication event. Formally, each *communication readiness* has the form of $\gamma.ch?$ or $\gamma.ch!$, where $\gamma \in T\Sigma^*_{\leq}$. We denote by RDY the set of communication readiness in the sequel.

Finally, we introduce two system variables, rdy and tr, to represent the ready set of communication events and the timed trace accumulated at the considered time, respectively. In what follows, we use $Var(P)$ to represent the set of process variables of P, plus the system variables $\{rdy, tr, now\}$ introduced above, which take values respectively from $\mathbb{R} \cup RDY \cup T\Sigma^*_{\leq} \cup \mathbb{R}^+$, denoted by Val.

States and Functions. To interpret a process $P \in Proc$, we define a state ds as a mapping from $Var(P)$ to Val, and denote by $\mathcal{D}$ the set of such states. Because of stochasticity, we introduce a random variable $\rho : \Omega \rightarrow \mathcal{D}$ to describe a distribution of all possible states. In addition,we introduce a stochastic process $H : Intv \times \Omega \rightarrow \mathcal{D}$ to represent the continuous flow of process P over the time interval $Intv$, i.e., state distributions on the interval. In what follows, we will abuse state distribution as state if not stated otherwise.

Given two states ρ_1 and ρ_2, we say ρ_1 and ρ_2 are parallelable iff for each $\omega \in \Omega$, $Dom(\rho_1(\omega)) \cap Dom(\rho_2(\omega)) = \{rdy, tr, now\}$ and $\rho_1(\omega)(now) = \rho_2(\omega)(now)$. Given two parallelable states ρ_1 and ρ_2, paralleling them over $X \subseteq \Sigma$ results in a set of new states, denoted by $\rho_1 \uplus \rho_2$, any of which ρ is given by

$$\rho(\omega)(v) \overset{\text{def}}{=} \begin{cases} \rho_1(\omega)(v) & \text{if } v \in Dom(\rho_1(\omega)) \setminus Dom(\rho_2(\omega)), \\ \rho_2(\omega)(v) & \text{if } v \in Dom(\rho_2(\omega)) \setminus Dom(\rho_1(\omega)), \\ \rho_1(\omega)(now) & \text{if } v = now, \\ \gamma, \text{ where } \gamma \in \rho_1(\omega)(tr)\rho_2(\omega)(tr) & \text{if } v = tr, \\ \rho_1(\omega)(rdy) \cup \rho_2(\omega)(rdy) & \text{if } v = rdy. \end{cases}$$

It makes no sense to distinguish any two states in $\rho_1 \uplus \rho_2$, so hereafter we abuse $\rho_1 \uplus \rho_2$ to represent any of its elements.$\rho_1 \uplus \rho_2$ will be used to represent states of parallel processes.

Given a random variable ρ, the update $\rho[v \rightarrow e]$ represents a new random variable such that for any $\omega \in \Omega$ and $x \in Var$, $\rho[v \rightarrow e](\omega)(x)$ is defined as the value of e if x is v, and $\rho(\omega)(x)$ otherwise. Given a stochastic process $X : [0, d) \times \Omega \rightarrow R^{d(s)}$, for any t in the domain, $\rho[s \rightarrow X_t]$ is a new random variable such that for any $\omega \in \Omega$ and $x \in Var$, $\rho[s \rightarrow X_t](\omega)(x)$ is defined as $X(t, w)$ if x is s, and $\rho(\omega)(x)$ otherwise.

At last, we define H^ρ_d as the stochastic process over interval $[\rho(now), \rho(now)+ d]$ such that for any $t \in [\rho(now), \rho(now) + d]$ and any ω, $H^\rho_d(t, \omega) = \rho[now \mapsto t](\omega)$, and moreover, $H^{\rho,s,X}_d$ as the stochastic process over interval $[\rho(now), \rho(now) + d]$ such that for any $t \in [\rho(now), \rho(now) + d]$ and any ω, $H^{\rho,s,X}_d(t, \omega) = \rho[now \mapsto t, rdy \mapsto \emptyset, s \mapsto X_t](\omega)$.

4.1 Operational Semantics

Each transition relation has the form of $(P, \rho) \xrightarrow{\alpha} (P', \rho', H)$, where P and P' are processes, α is an event, ρ, ρ' are states, H is a stochastic process. It expresses that starting from initial state ρ, P evolves into P' by performing event α, and ends in state ρ' and the execution history of α is recorded by continuous flow H. When the transition is discrete and thus produces a flow on a point interval (i.e. current time now), we will write $(P, \rho) \xrightarrow{\alpha} (P', \rho')$ instead of $(P, \rho) \xrightarrow{\alpha} (P', \rho', \{\rho(now) \mapsto \rho'\})$. The label α represents events, which can be an internal event like skip, assignment, or a termination of a continuous *etc*, uniformly denoted by τ, or an external communication event $ch!c$ or $ch?c$, or an internal communication $ch.c$, or a time delay d that is a positive real number. We call the events but the time delay *discrete events*, and will use β to range over them. We define the dual of $ch?c$ (denoted by $\overline{ch?c}$) as $ch!c$, and vice versa, and define $comm(ch!c, ch?c)$ or $comm(ch?c, ch!c)$ as the communication $ch.c$. In the operational semantics, besides the timed communications, we will also record the internal events that have occurred till now in tr.

For page limit, we present the semantics for the new constructs of SHCSP in the paper in Table 1. The semantics for the rest is same to HCSP, which can be found at [17]. The semantics for probabilistic choice is given by rules (PCho-1) and (PCho-2): it is defined with respect to a random variable U which distributes uniformly in $[0, 1]$, such that for any sample ω, if $U(\omega) \leq p$, then P is taken, otherwise, Q is taken. In either case, it is assumed that an internal action happened. A stochastic dynamics can continuously evolve for d time units if B always holds during this period, see (Cont-1). In (Cont-1), the variable X solves the stochastic process and the ready set keeps unchanged, reflected by the flow $H_d^{\rho,s,X}$. The stochastic dynamics terminates at a point whenever B turns out false at a neighborhood of the point (Cont-2). Communication interrupt evolves for d time units if none of the communications ch_i* is ready (IntP-1), or is interrupted to execute $ch_{i_j}*$ whenever $ch_{i_j}*$ occurs first (IntP-2), or terminates immediately in case the continuous terminates before any communication happening (IntP-3).

The following theorem indicates that the semantics of SHCSP is well defined.

Theorem 1. *For each transition $(P, \rho) \xrightarrow{\alpha} (P', \rho', H)$, H is an almost surely càdlàg process and adapted to the completed filtration $(\mathcal{F}_t)_{t \geq 0}$ (generated by ρ, the Brownian motion $(B_s)_{s \leq t}$, the weights $\{\omega_i\}_{i \in I}$ and uniform U process) and the evolving time from P to P', denoted by $\Delta(P, P')$, is a Markov time.*

Proof. The proof of this theorem can be found at [17].

5 Assertions and Specifications

In this section, we define a specification logic for reasoning about SHCSP programs. We will first present the assertions including syntax and semantics, and then the specifications based on Hoare triples. The proof system will be given in next section.

5.1 Assertion Language

The assertion language is essentially defined by a first-order logic with emphasis on the notion of explicit time and the addition of several specific predicates on occurrence of communication traces and events. Before giving the syntax of assertions, we introduce three kinds of expressions first.

$$\begin{array}{l}
h \ ::= \varepsilon \mid \langle ch.E, T\rangle \mid h \cdot h \mid h^* \\
E ::= c \mid x \mid f^k(E_1, ..., E_k) \\
T \ ::= o \mid now \mid u^l(T_1, ..., T_l)
\end{array}$$

h defines trace expressions, among which $\langle ch.E, T\rangle$ represents that there is a value E transmitted along channel ch at time T. E defines value expressions,

Table 1. The semantics of new constructs of SHCSP

$$\frac{U \text{ is a random variable distributed uniformly in } [0,1],\ U(\omega) \le p}{(P \sqcup_p Q, \rho) \xrightarrow{\tau} \big(P,\ \rho[tr \mapsto tr \cdot \langle \tau, now\rangle]\big)} \qquad \text{(PCho-1)}$$

$$\frac{U \text{ is a random variable distributed uniformly in } [0,1],\ U(\omega) > p}{(P \sqcup_p Q, \rho) \xrightarrow{\tau} \big(Q,\ \rho[tr \mapsto tr \cdot \langle \tau, now\rangle]\big)} \qquad \text{(PCho-2)}$$

$$\frac{\begin{array}{l} X : [0,d) \times \Omega \to \mathbb{R}^{d(s)} \text{ is the solution of} \\ ds = bdt + \sigma dW \wedge \forall t \in [0,d), \forall \omega.\rho[now \mapsto now + t, s \mapsto X_t](\omega)(B) = \mathbf{T} \end{array}}{(\langle ds = bdt + \sigma dW \& B\rangle, \rho) \xrightarrow{d} \begin{pmatrix} \langle ds = bdt + \sigma dW \& B\rangle, \\ \rho[now \mapsto now + d, s \mapsto X_d], H_d^{\rho,s,X} \end{pmatrix}} \qquad \text{(Cont-1)}$$

$$\frac{\begin{array}{l} \exists \omega.(\rho(\omega)(B) = \mathbf{F}) \text{ or } (X : [0,d) \times \Omega \to \mathbb{R}^{d(s)} \text{ is the solution of } ds = bdt + \sigma dW, \\ \exists \varepsilon > 0 \forall t \in (0,\varepsilon) \exists \omega.\rho[now \mapsto now + t, s \mapsto X_t](\omega)(B) = \mathbf{F}) \end{array}}{(\langle ds = bdt + \sigma dW \& B\rangle, \rho) \xrightarrow{\tau} (\epsilon, \rho[tr \mapsto tr \cdot \langle \tau, now\rangle)} \qquad \text{(Cont-2)}$$

$$\frac{\begin{array}{l} (ch_i*; Q_i, \rho) \xrightarrow{d} (ch_i*; Q_i, \rho_i', H_i), \quad \forall i \in I \\ (\langle ds = bdt + \sigma dW \& B\rangle, \rho) \xrightarrow{d} (\langle ds = bdt + \sigma dW \& B\rangle, \rho', H) \end{array}}{\begin{array}{l} (\langle ds = bdt + \sigma dW \& B\rangle \trianglerighteq []_{i\in I}(\omega_i \cdot ch_i* \to Q_i), \rho) \xrightarrow{d} \\ \qquad \begin{pmatrix} \langle ds = bdt + \sigma dW \& B\rangle \trianglerighteq []_{i\in I}(\omega_i \cdot ch_i* \to Q_i), \\ \rho'[rdy \mapsto \cup_{i\in I}\rho_i'(rdy)], H[rdy \mapsto \cup_{i\in I}\rho_i'(rdy)] \end{pmatrix} \end{array}} \qquad \text{(IntP-1)}$$

$$\frac{\begin{array}{l} \{\overline{ch_{i_k}}*\}_{1\le k\le n} \text{ get ready simultaneously while others not} \\ U \text{ is a random variable distributed uniformly in [0,1], and for } 1 \le j \le n \\ \frac{\sum_{k=1}^{j-1}\omega_{i_k}}{\sum_{k=1}^{n}\omega_{i_k}} \le U(\omega) < \frac{\sum_{k=1}^{j}\omega_{i_k}}{\sum_{k=1}^{n}\omega_{i_k}} \text{ and } (ch_{i_j}*; Q_{i_j}, \rho) \xrightarrow{ch_{i_j}*} (Q_{i_j}, \rho') \end{array}}{(\langle ds = bdt + \sigma dW \& B\rangle \trianglerighteq []_{i\in I}(\omega_i \cdot ch_i* \to Q_i), \rho) \xrightarrow{ch_{i_j}*} (Q_{i_j}, \rho')} \qquad \text{(IntP-2)}$$

$$\frac{(\langle ds = bdt + \sigma dW \& B\rangle, \rho) \xrightarrow{\tau} (\epsilon, \rho')}{(\langle ds = bdt + \sigma dW \& B\rangle \trianglerighteq []_{i\in I}(\omega_i \cdot ch_i* \to Q_i), \rho) \xrightarrow{\tau} (\epsilon, \rho')} \qquad \text{(IntP-3)}$$

including a value constant c, a variable x, or arithmetic value expressions. T defines time expressions, including a time constant o, system variable now, or arithmetic time expressions.

The categories of the assertion language include terms, denoted by θ, θ_1 *etc.*, state formulas, denoted by S, S_1 *etc.*, formulas, denoted by φ, φ_1 *etc.*, and probability formulas, denoted by $\mathcal{P}$ *etc.*, which are given by the following BNFs:

$$\begin{aligned}
\theta &::= E \mid T \mid h \mid tr \\
S &::= \bot \mid R^n(\theta_1, ..., \theta_n) \mid h.ch? \mid h.ch! \mid \neg S \mid S_1 \vee S_2 \\
\varphi &::= \bot \mid S \ \mathsf{at}\ T \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \forall v.\varphi \mid \forall t.\varphi \\
\mathcal{P} &::= P(\varphi) \bowtie p \mid \neg\mathcal{P} \mid \mathcal{P} \vee \mathcal{P}
\end{aligned}$$

The terms θ include value, time and trace expressions, plus trace variable tr. The state expressions S include false (denoted by $\bot$), truth-valued relation R^n on terms, readiness, and logical combinations of state formulas. In particular, the readiness $h.ch?$ or $h.ch!$ represents that the communication event $ch?$ or $ch!$ is enabled, and prior to it, the sequence of communications recorded in h has occurred. The formulas φ include false, a primitive S at T representing that S holds at time T; and logical combinations of formulas (v, t represent logical variables for values and time resp.). For time primitive, we have an axiom that $(S_1\ \mathsf{at}\ T \wedge S_2\ \mathsf{at}\ T) \Leftrightarrow (S_1 \wedge S_2)\ \mathsf{at}\ T$. We omit all the other axiom and inference rules for the formulas, that are same to first-order logic. The probability formula $\mathcal{P}$ has the form $P(\varphi) \bowtie p$, where $\bowtie \in \{<, \leq, >, \geq\}$, $p \in \mathbb{Q} \cap [0, 1]$, or the logical composition of probability formulas free of quantifiers. In particular, $P(\varphi) \bowtie p$ means that φ is true with probability $\bowtie p$. For the special case $P(\varphi) = 1$, we write φ for short.

In the sequel, we use the standard logical abbreviations, as well as

$$\varphi\ \mathsf{dr}\ [T_1, T_2] \stackrel{\text{def}}{=} \forall t.(T_1 \leq t \leq T_2) \Rightarrow \varphi\ \mathsf{at}\ t$$
$$\varphi\ \mathsf{in}\ [T_1, T_2] \stackrel{\text{def}}{=} \exists t.(T_1 \leq t \leq T_2) \wedge \varphi\ \mathsf{at}\ t$$

Interpretation. In the following, we will use a random variable $Z : \Omega \rightarrow (Var \rightarrow Val)$ to describe the current state and a stochastic process $\mathcal{H} : [0, +\infty) \times \Omega \rightarrow (Var \rightarrow Val)$ to represent the whole evolution. The semantics of a term θ is a function $[\![\theta]\!] : (\Omega \rightarrow (Var \rightarrow Val)) \rightarrow (\Omega \rightarrow Val)$ that maps any random variable Z to a random variable $[\![\theta]\!]^Z$, defined as follows:

$$\begin{aligned}
&[\![c]\!]^Z = c \\
&[\![x]\!]^Z = Y \text{ where } Y(\omega) = Z(\omega)(x) \text{ for } \omega \in \Omega \\
&[\![f^k(E_1, ..., E_k)]\!]^Z = f^k([\![E_1]\!]^Z, ..., [\![E_k]\!]^Z) \\
&[\![o]\!]^Z = o \\
&[\![now]\!]^Z = Y \text{ where } Y(\omega) = Z(\omega)(now) \text{ for } \omega \in \Omega \\
&[\![u^l(T_1, ..., T_l)]\!]^Z = u^l([\![T_1]\!]^Z, ..., [\![T_l]\!]^Z) \\
&[\![\varepsilon]\!]^Z = \varepsilon \\
&[\![\langle ch.E, T\rangle]\!]^Z = \langle ch.[\![E]\!]^Z, [\![T]\!]^Z\rangle \\
&[\![h_1 \cdot h_2]\!]^Z = [\![h_1]\!]^Z \cdot [\![h_2]\!]^Z \\
&[\![h^*]\!]^Z = ([\![h]\!]^Z)^*
\end{aligned}$$

The semantics of state formula S is a function $\llbracket S \rrbracket : (\Omega \rightarrow (\mathit{Var} \rightarrow \mathit{Val})) \rightarrow (\Omega \rightarrow \{0,1\})$ that maps any random variable Z describing the current state to a boolean random variable $\llbracket S \rrbracket^Z$, defined as follows:

$$
\begin{array}{l}
\llbracket \bot \rrbracket^Z = 0 \\
\llbracket R^n(\theta_1, \ldots, \theta_n) \rrbracket^Z = R^n(\llbracket \theta_1 \rrbracket^Z, \ldots, \llbracket \theta_n \rrbracket^Z) \\
\text{where } R^n(\llbracket \theta_1 \rrbracket^Z, \ldots, \llbracket \theta_n \rrbracket^Z)(\omega) = R^n(\llbracket \theta_1 \rrbracket^Z(\omega), \ldots, \llbracket \theta_n \rrbracket^Z(\omega)) \\
\llbracket h.ch? \rrbracket^Z = \mathcal{I}_{\{\omega \in \Omega | \llbracket h \rrbracket^Z(\omega).ch? \in Z(\omega)(rdy)\}} \\
\llbracket h.ch! \rrbracket^Z = \mathcal{I}_{\{\omega \in \Omega | \llbracket h \rrbracket^Z(\omega).ch! \in Z(\omega)(rdy)\}} \\
\llbracket \neg S \rrbracket^Z = 1 - \llbracket S \rrbracket^Z \\
\llbracket S_1 \vee S_2 \rrbracket^Z = \llbracket S_1 \rrbracket^Z + \llbracket S_2 \rrbracket^Z - \llbracket S_1 \rrbracket^Z * \llbracket S_2 \rrbracket^Z
\end{array}
$$

where given a set S, the characteristic function $\mathcal{I}_S$ is defined such that $\mathcal{I}_S(w) = 1$ if $w \in S$ and $\mathcal{I}_S(w) = 0$ otherwise. The semantics of formula φ is interpreted over a stochastic process and an initial random variable. More precisely, it's a function $\llbracket \varphi \rrbracket : ([0, +\infty) \times \Omega \rightarrow (\mathit{Var} \rightarrow \mathit{Val})) \rightarrow (\Omega \rightarrow (\mathit{Var} \rightarrow \mathit{Val})) \rightarrow (\Omega \rightarrow \{0,1\})$ that maps a stochastic process $\mathcal{H}$ with initial state Z to a boolean random variable $\llbracket \varphi \rrbracket^{\mathcal{H},Z}$. The definition is given below:

$$
\begin{array}{l}
\llbracket \bot \rrbracket^{\mathcal{H},Z} = 0 \\
\llbracket S \ \mathsf{at} \ T \rrbracket^{\mathcal{H},Z} = \llbracket S \rrbracket^{\mathcal{H}(\llbracket T \rrbracket^Z)} \\
\llbracket \neg \varphi \rrbracket^{\mathcal{H},Z} = 1 - \llbracket \varphi \rrbracket^{\mathcal{H},Z} \\
\llbracket \varphi_1 \vee \varphi_2 \rrbracket^{\mathcal{H},Z} = \llbracket \varphi_1 \rrbracket^{\mathcal{H},Z} + \llbracket \varphi_2 \rrbracket^{\mathcal{H},Z} - \llbracket \varphi_1 \rrbracket^{\mathcal{H},Z} * \llbracket \varphi_2 \rrbracket^{\mathcal{H},Z} \\
\llbracket \forall v.\varphi \rrbracket^{\mathcal{H},Z} = \inf\{\llbracket \varphi[b/v] \rrbracket^{\mathcal{H},Z} : b \in \mathbb{R}\} \\
\llbracket \forall t.\varphi \rrbracket^{\mathcal{H},Z} = \inf\{\llbracket \varphi[b/t] \rrbracket^{\mathcal{H},Z} : b \in \mathbb{R}^+\}
\end{array}
$$

The semantics of probability formula $\mathcal{P}$ is defined by function $\llbracket \mathcal{P} \rrbracket : ([0, +\infty) \times \Omega \rightarrow (\mathit{Var} \rightarrow \mathit{Val})) \rightarrow (\Omega \rightarrow (\mathit{Var} \rightarrow \mathit{Val})) \rightarrow \{0,1\}$ that maps a stochastic process $\mathcal{H}$ with initial state Z to a boolean variable $\llbracket \mathcal{P} \rrbracket^{\mathcal{H},Z}$. Formally,

$$
\llbracket P(\varphi) \bowtie p \rrbracket^{\mathcal{H},Z} = (P(\llbracket \varphi \rrbracket^{\mathcal{H},Z} = 1) = P(\{\omega \in \Omega : \llbracket \varphi \rrbracket^{\mathcal{H},Z}(\omega) = 1\}) \bowtie p)
$$

The semantics for $\neg$ and $\vee$ can be defined as usual.

We have proved that the terms and formulas of the assertion language are measurable, stated by the following theorem:

Theorem 2 (Measurability). *For any random variable Z and any stochastic process $\mathcal{H}$, the semantics of $\llbracket \theta \rrbracket^Z$, $\llbracket S \rrbracket^Z$ and $\llbracket \varphi \rrbracket^{\mathcal{H},Z}$ are random variables (i.e. measurable).*

Proof. The proof of this theorem can be found at [17].

5.2 Specifications

Based on the assertion language, the specification for a SHCSP process P is defined as a Hoare triple of the form $\{A; E\}\, P\, \{R; C\}$, where A, E, R, C are probability formulas. A and R are *precondition* and *postcondition*, which specify the initial state and the terminating state of P respectively. For both of them, the formulas φ occurring in them have the special form $S\ \mathsf{at}\ \mathit{now}$, and we will write S for short. E is called an *assumption* of P, which expresses the timed occurrence of the dual of communication events provided by the environment. C is called a *commitment* of P, which expresses the timed occurrence of communication events, and the real-time properties of P.

Definition 1 (Validity). *We say a Hoare triple* $\{A; E\}\, P\, \{R; C\}$ *is* valid*, denoted by* $\models \{A; E\}\, P\, \{R; C\}$*, iff for any process* Q*, any initial states* ρ_1 *and* ρ_2*, if* P *terminates, i.e.*$(P \| Q, \rho_1 \uplus \rho_2) \xrightarrow{\alpha^*} (\epsilon \| Q', \rho_1' \uplus \rho_2', \mathcal{H})$ *then* $[\![A]\!]^{\rho_1}$ *and* $[\![E]\!]^{\mathcal{H},\rho_2}$ *imply* $[\![R]\!]^{\rho_1'}$ *and* $[\![C]\!]^{\mathcal{H},\rho_1'}$*, where* $\mathcal{H}$ *is the stochastic process of the evolution.*

6 Proof System

We present a proof system for reasoning about all valid Hoare triples for SHCSP processes. First we axiomatize SHCSP language by defining the axioms and inference rules for all the primitive and compound constructs, and then the general rules and axioms that are applicable to all processes.

Skip. The rule for skip is very simple. Indicated by $\top$, the skip process requires nothing from the environment for it to execute, and guarantees nothing during its execution.

$$\{A; \top\}\, \mathbf{skip}\, \{A; \top\}$$

Assignment. The assignment $x := e$ changes nothing but assigns x to e in the final state, taking no time to complete.

$$\{A[e/x]; \top\}\, x := e\, \{A; \top\}$$

Input. For input $ch?x$, we use logical variables o to denote the starting time, h the initial trace, and v the initial value of x respectively, in the precondition. The assumption indicates that the compatible output event is not ready during $[o, o_1)$, and at time o_1, it becomes ready. As a consequence of the assumption, during the whole interval $[o, o_1]$, the input event keeps waiting and ready, as indicated by the commitment. At time o_1, the communication occurs and terminates immediately. As indicated by the postcondition, x is assigned by some value v' received, the trace is augmented by the new pair $\langle ch.v', o_1\rangle$, and now is increased to o_1. Assume A does not contain tr and o_1 is finite (and this assumption will be adopted for the rest of the paper). Let h' be $h[v/x, o/\mathit{now}] \cdot \langle ch.v', o_1\rangle$, the rule is presented as follows:

$$\begin{gathered}\{A \wedge \mathit{now} = o \wedge \mathit{tr} = h \wedge x = v; \neg h.ch!\ \mathsf{dr}\ [o, o_1) \wedge h.ch!\ \mathsf{at}\ o_1\} ch?x \\ \{A[o/\mathit{now}] \wedge \mathit{now} = o_1 \wedge \exists v'.(x = v' \wedge \mathit{tr} = h'); h.ch?\ \mathsf{dr}\ [o, o_1]\}\end{gathered}$$

A communication event is equivalent to a sequential composition of a wait statement and an assignment, both of which are deterministic. Thus, as shown above, the formulas related to traces and readiness hold with probability 1.

If such finite o_1 does not exist, i.e., the compatible output event will never become available. As a consequence, the input event will keep waiting forever, as shown by the following rule:

$$\{A \wedge now = o \wedge tr = h; \neg h.ch! \ \mathsf{dr}\ [o, \infty)\} ch?x$$
$$\{A[o/now] \wedge now = \infty; h.ch? \ \mathsf{dr}\ [o, \infty)\}$$

Output. Similarly, for output $ch!e$, we have one rule for the case when the compatible input event becomes ready in finite time. Thus the communication occurs successfully.

$$\{A \wedge now = o \wedge tr = h; \neg h.ch? \ \mathsf{dr}\ [o, o_1) \wedge h.ch? \ \mathsf{at}\ o_1\} ch!e$$
$$\{A[o/now] \wedge now = o_1 \wedge tr = h[o/now] \cdot \langle ch.e, o_1 \rangle, h.ch! \ \mathsf{dr}\ [o, o_1]\}$$

We also have another rule for the case when the compatible input event will never get ready.

$$\{A \wedge now = o \wedge tr = h; (\neg h.ch?) \ \mathsf{dr}\ [o, \infty)\}\ ch!e$$
$$\{A[o/now] \wedge now = \infty; h.ch! \ \mathsf{dr}\ [o, \infty)\}$$

Stochastic Differential Equation. Let f be a function, and $\lambda > 0, p \geq 0$ are real values. We have the following rule for $\langle ds = bdt + \sigma dW \& B \rangle$.

$$\frac{\begin{array}{c} f(s) \in C^2(\mathbb{R}^n, \mathbb{R}) \text{ has compact support on } B, \lambda, p > 0 \text{ and} \\ A \to B \to (f \leq \lambda p) \quad B \to (f \geq 0) \wedge (Lf \leq 0) \end{array}}{\begin{array}{c} \{A \wedge s = s_0 \wedge now = o; \top\} \langle ds = bdt + \sigma dW \& B \rangle \{P(f(s) \geq \lambda) \leq p \wedge A[s_0/s, o/now] \\ \wedge now = o + d \wedge cl(B); B \wedge P(f(s) \geq \lambda \ \mathsf{dr}\ [o, o+d]) \leq p\} \end{array}}$$

where o, s_0 are logical variables denoting the starting time and the initial value of s resp., d is the execution time of the SDE, and $cl(B)$ returns the closure of B, e.g. $cl(x < 2) = x \leq 2$; and the Lie derivative $Lf(s)$ is defined as $\sum\limits_i b_i(s) \frac{\partial f}{\partial s_i}(s) + \frac{1}{2} \sum\limits_{i,j} (\sigma(s)\sigma(s)^T)_{i,j} \frac{\partial^2 f}{\partial s_i \partial s_j}(s)$. The rule states that, if the initial state of the SDE satisfies $f \leq \lambda p$, and in the domain B, f is always non-negative and Lf is non-positive, then during the whole evolution of the SDE, the probability of $f(s) \geq \lambda$ is less than or equal to p; on the other hand, during the evolution, the domain B holds almost surely, while at the end, the closure of B holds almost surely.

Sequential Composition. For $P; Q$, we use o to denote the starting time, and o_1 the termination time of P, if P terminates, which is also the starting time of Q. The first rule is for the case when P terminates.

$$\frac{\{A \wedge now = o; E\}\ P\ \{R_1 \wedge now = o_1; C_1\} \quad \{R_1 \wedge now = o_1; C_1\}\ Q\ \{R; C\}}{\{A; E\}\ P; Q\ \{R; C\}}$$

On the other hand, if P does not terminate, the effect of executing $P; Q$ is same to that of executing P itself.

$$\frac{\{A \wedge now = o; E\}\ P\ \{R \wedge now = \infty; C\}}{\{A \wedge now = o; E\}\ P; Q\ \{R \wedge now = \infty; C\}}$$

Conditional. There are two rules depending on whether B holds or not initially.

$$\frac{A \Rightarrow B \quad \{A; E\}\, P\, \{R; C\}}{\{A; E\}\, B \rightarrow P\, \{R; C\}} \text{ and } \frac{A \Rightarrow \neg B}{\{A; \top\}\, B \rightarrow P\, \{A; \top\}}$$

Probabilistic Choice. The rule for $P \sqcup_p Q$ is defined as follows:

$$\frac{\begin{array}{c}\{A \wedge now = o; E\}\, P\, \{P(S) \bowtie_1 p_1; P(\varphi) \bowtie_2 p_2\} \\ \{A \wedge now = o; E\}\, Q\, \{P(S) \bowtie_1 q_1; P(\varphi) \bowtie_2 q_2\}\end{array}}{\{A \wedge now = o; E\}\ \ P \sqcup_p Q\ \ \{P(S) \bowtie_1 pp_1 + (1-p)q_1; P(\varphi) \bowtie_2 pp_2 + (1-p)q_2\}}$$

where $\bowtie_1, \bowtie_2$ are two relational operators. The final postcondition indicates that, if after P executes S holds with probability $\bowtie_1 p_1$, and after Q executes S holds with probability $\bowtie_1 q_1$, then after $P \sqcup_p Q$ executes, S holds with probability $\bowtie_1 pp_1 + (1-p)q_1$; The history formula can be understood similarly.

Communication Interrupt. We define the rule for the special case $\langle ds = bdt + \sigma dW \& B\rangle \trianglerighteq (ch?x \rightarrow Q)$ for simplicity, which can be generalized to general case without any difficulty. We use o_F to denote the execution time of the SDE. The premise of the first rule indicates that the compatible event (i.e. $h.ch!$) is not ready after the continuous terminates. For this case, the effect of executing the whole process is thus equivalent to that of executing the SDE.

$$\frac{\begin{array}{c}\{A \wedge now = o; E\}\langle ds = bdt + \sigma dW \& B\rangle\{R \wedge now = o + o_F; C\} \\ A \wedge now = o \wedge E \Rightarrow (tr = h \wedge \neg h.ch!\ \mathsf{dr}\ [o, o + o_F])\end{array}}{\{A \wedge now = o; E\}\ \langle ds = bdt + \sigma dW \& B\rangle \trianglerighteq (ch?x \rightarrow Q)\ \{R \wedge now = o + o_F; C\}}$$

In contrary, when the compatible event gets ready before the continuous terminates, the continuous will be interrupted by the communication, which is then followed by Q. Thus, as shown in the following rule, the effect of executing the whole process is equivalent to that of executing $ch?x; Q$, plus that of executing the SDE before the communication occurs, i.e. in the first o_1 time units.

$$\frac{\begin{array}{c}\{A \wedge now = o; E\}\langle ds = bdt + \sigma dW \& B\rangle\{R \wedge now = o + o_F; C\} \\ (A \wedge now = o \wedge E) \Rightarrow (tr = h \wedge h.ch!\ \mathsf{at}\ (o + o_1) \wedge o_1 \leq o_F) \\ \{A \wedge B \wedge now = o; E\}\, ch?x; Q\, \{R_1; C_1\}\end{array}}{\begin{array}{c}\{A \wedge now = o; E\}\ \langle ds = bdt + \sigma dW \& B\rangle \trianglerighteq (ch?x \rightarrow Q) \\ \qquad\qquad\qquad\qquad \{R_1; R|_{[o, o+o_1)} \wedge C_1\}\end{array}}$$

where $R|_{[o,o+o_1]}$ extracts from R the formulas before $o + o_1$, e.g., $(P(S\ \mathsf{at}\ T) \bowtie p)|_{[o,o+o_1]}$ is equal to $P(S\ \mathsf{at}\ T) \bowtie p$ if T is less or equal to $o + o_1$, and true otherwise.

Parallel Composition

For $P\|Q$, let X be $X_1 \cap X_2$ where $X_1 = \Sigma(P)$ and $X_2 = \Sigma(Q)$, then

$$\frac{\begin{array}{c}A \Rightarrow A_1 \wedge A_2, \quad \{A_1 \wedge now = o; E_1\}\, P\, \{R_1 \wedge tr = \gamma_1 \wedge now = o_1; C_1\} \\ \{A_2 \wedge now = o; E_2\}\, Q\, \{R_2 \wedge tr = \gamma_2 \wedge now = o_2; C_2\} \\ \forall ch \in X.(C_1[o_1/now]\restriction_{ch} \Rightarrow E_2\restriction_{ch}) \wedge (C_2[o_2/now]\restriction_{ch} \Rightarrow E_1\restriction_{ch}) \\ \forall dh \in X_1 \setminus X.E\restriction_{dh} \Rightarrow E_1\restriction_{dh} \quad \forall dh' \in X_2 \setminus X.E\restriction_{dh'} \Rightarrow E_2\restriction_{dh'}\end{array}}{\{A \wedge now = o; E\}\, P\|Q\, \{R; C_1' \wedge C_2'\}}$$

where A_1 is a property of P (i.e., it only contains variables of P), A_2 a property of Q, and o_1 and o_2, γ_1 and γ_2 logical variables representing the time and trace at termination of P and Q respectively. Let o_m be $\max\{o_1, o_2\}$, R, C_1' and C_2' are defined as follows:

$$R \overset{\text{def}}{=} R_1[\gamma_1/tr, o_1/now] \wedge R_2[\gamma_2/tr, o_2/now] \wedge now = o_m \wedge \gamma_1 \upharpoonright_X = \gamma_2 \upharpoonright_X \wedge tr = \gamma_1\gamma_2$$
$$C_i' \overset{\text{def}}{=} C_i[o_i/now] \wedge R_i'[o_i/now] \ \mathsf{dr}\ [o_i, o_m) \text{ for } i = 1, 2$$

where for $i = 1, 2$, $R_i \Rightarrow R_i'$ but $tr \notin R_i'$. At termination of $P \| Q$, the time will be the maximum of o_1 and o_2, and the trace will be the alphabetized parallel of the traces of P and Q, i.e. γ_1, γ_2. In C_1' and C_2', we specify that none of variables of P and Q except for now and tr will change after their termination.

Repetition. For P^*, let k be an arbitrary non-negative integer, then ($tr \notin A$)

$$\frac{\begin{array}{c}\{A \wedge now = o + k * t \wedge tr = (h \cdot \alpha^k); E[o/now]\}\ P \\ \{A \wedge now = o + (k+1) * t \wedge tr = (h \cdot \alpha^{k+1}); C\}\end{array}}{\{A \wedge now = o \wedge tr = h; E\}\ P^*\ \{A \wedge now = o' \wedge tr = (h \cdot \alpha^*) + \tau; C \vee (o = o' \ \mathsf{at}\ now)\}}$$

t and α are logical variables representing the time elapsed and trace accumulated respectively by each execution of P, and o and o' denote the starting and termination time of the loop (o' could be infinite).

The general rules that are applicable to all processes, such as Monotonicity, Case Analysis, and so on, are similar to the traditional Hoare Logic. We will not list them here for page limit.

Theorem 3 (Soundness). *If* $\vdash \{A; E\}\, P\, \{R; C\}$, *then* $\models \{A; E\}\, P\, \{R; C\}$, *i.e. every theorem of the proof system is valid.*

Proof. The proof of this theorem can be found at [17].

Example 1. For the aircraft example, define $f(x, y)$ as $|y|$, assume $f(x_s, y_0) = |y_0| \leq \lambda p$, where $p \in [0, 1]$. Obviously, $B \to (f \geq 0) \wedge (Lf \leq 0)$ holds. By applying the inference rule of SDE, we have the following result:

$$\{now = o; True\}\ P_{Air}\ \{\begin{array}{l}\exists d. now = o + d \wedge B \wedge P(f \geq \lambda) \leq p; \\ B \wedge P(f \geq \lambda \ \mathsf{dr}\ [o, o+d]) \leq p\end{array}\}$$

which shows that, the probability of the aircraft entering the dangerous state is always less than or equal to p during the flight. Thus, to guarantee the safety of the aircraft, p should be as little as possible. For instance, if the safety factor of the aircraft is required to be 99.98%, then p should be less than or equal to 0.0002, and in correspondence, $|y_0| \leq \frac{\lambda}{5000}$ should be satisfied.

7 Conclusion

This paper presents stochastic HCSP (SHCSP) for modelling hybrid systems with probability and stochasticity. SHCSP is expressive but complicated with interacting discrete, continuous and stochastic dynamics. We have defined the semantics of stochastic HCSP and proved that it is well-defined with respect to stochasticity. We propose an assertion language for specifying time-related and probability-related properties of SHCSP, and have proved the measurability of it. Based on the assertion language, we define a compositional Hoare Logic for specifying and verifying SHCSP processes. The logic is an extension of traditional Hoare Logic, and can be used to reason about how the probability of a property changes with respect to the execution of a process. To illustrate our approach, we model and verify a case study on a flight planing problem at the end.

References

1. Abate, A., Prandini, M., Lygeros, J., Sastry, S.: Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. Automatica **44**(11), 2724–2734 (2008)
2. Altman, E., Gaitsgory, V.: Asymptotic optimization of a nonlinear hybrid system governed by a Markov decision process. SIAM Journal of Control and Optimization **35**(6), 2070–2085 (1997)
3. Bujorianu, M.L.: Extended stochastic hybrid systems and their reachability problem. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 234–249. Springer, Heidelberg (2004)
4. Bujorianu, M.L., Lygeros, J.: Toward a general theory of stochastic hybrid systems. In: Blom, H.A.P., Lygeros, J. (eds.) Stochastic Hybrid Systems. LNCIS, vol. 337, pp. 3–30. Springer, Heidelberg (2006)
5. Bujorianu, M.L., Lygeros, J., Bujorianu, M.C.: Bisimulation for general stochastic hybrid systems. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 198–214. Springer, Heidelberg (2005)
6. Fränzle, M., Hahn, E.M., Hermanns, H., Wolovick, N., Zhang, L.: Measurability and safety verification for stochastic hybrid systems. In: HSCC 2011, pp. 43–52. ACM (2011)
7. Hahn, E.M., Hartmanns, A., Hermanns, H., Katoen, J.: A compositional modelling and analysis framework for stochastic hybrid systems. Formal Methods in System Design **43**(2), 191–232 (2013)
8. Hahn, E.M., Hermanns, H., Wachter, B., Zhang, L.: PASS: abstraction refinement for infinite probabilistic models. In: Esparza, J., Majumdar, R. (eds.) TACAS 2010. LNCS, vol. 6015, pp. 353–357. Springer, Heidelberg (2010)
9. He, J.: From CSP to hybrid systems. In: A Classical Mind, Essays in Honour of C.A.R. Hoare, pp. 171–189. Prentice Hall International (UK) Ltd. (1994)
10. Henzinger, T.A.: The theory of hybrid automata. In: LICS 1996, pp. 278–292, July 1996
11. Hoare, C.A.R.: An axiomatic basis for computer programming. Commun. ACM **12**(10), 576–580 (1969)
12. Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall (1985)
13. Hu, J., Lygeros, J., Sastry, S.S.: Towards a theory of stochastic hybrid systems. In: Lynch, N.A., Krogh, B.H. (eds.) HSCC 2000. LNCS, vol. 1790, p. 160. Springer, Heidelberg (2000)
14. Liu, J., Lv, J., Quan, Z., Zhan, N., Zhao, H., Zhou, C., Zou, L.: A calculus for hybrid CSP. In: Ueda, K. (ed.) APLAS 2010. LNCS, vol. 6461, pp. 1–15. Springer, Heidelberg (2010)
15. Meseguer, J., Sharykin, R.: Specification and analysis of distributed object-based stochastic hybrid systems. In: Hespanha, J.P., Tiwari, A. (eds.) HSCC 2006. LNCS, vol. 3927, pp. 460–475. Springer, Heidelberg (2006)

16. Morgan, C., McIver, A., Seidel, K., Sanders, J.W.: Refinement-oriented probability for CSP. Formal Asp. Comput. **8**(6), 617–647 (1996)
17. Peng, Y., Wang, S., Zhan, N., Zhang, L.: Extending hybrid CSP with probability and stochasticity. Technical report, Institute of Software, Chinese Academy of Sciences (2015). http://arxiv.org/abs/1509.01660
18. Platzer, A.: Stochastic differential dynamic logic for stochastic hybrid programs. In: Bjørner, N., Sofronie-Stokkermans, V. (eds.) CADE 2011. LNCS, vol. 6803, pp. 446–460. Springer, Heidelberg (2011)
19. Prandini, M., Hu, J.: Application of reachability analysis for stochastic hybrid systems to aircraft conflict prediction. In: 47th IEEE Conference on Decision and Control (CDC), pp. 4036–4041. IEEE (2008)
20. Sproston, J.: Decidable model checking of probabilistic hybrid automata. In: Joseph, M. (ed.) FTRTFT 2000. LNCS, vol. 1926, p. 31. Springer, Heidelberg (2000)
21. Zhang, L., She, Z., Ratschan, S., Hermanns, H., Hahn, E.M.: Safety verification for probabilistic hybrid systems. In: Touili, T., Cook, B., Jackson, P. (eds.) CAV 2010. LNCS, vol. 6174, pp. 196–211. Springer, Heidelberg (2010)
22. Zhou, C., Wang, J., Ravn, A.P.: A formal description of hybrid systems. In: Alur, R., Sontag, E.D., Henzinger, T.A. (eds.) HS 1995. LNCS, vol. 1066. Springer, Heidelberg (1996)
23. Zuliani, P., Platzer, A., Clarke, E.M.: Bayesian statistical model checking with application to stateflow/simulink verification. Formal Methods in System Design **43**(2), 338–367 (2013)