A Probabilistic Calculus of Cyber-Physical Systems[☆,☆☆]

Ruggero Lanotte[a], Massimo Merro[b,*], Simone Tini[a]

[a] *Dipartimento di Scienza e Alta Tecnologia, Università degli Studi dell'Insubria, Via Valleggio 11, 22100 Como, Italy*
[b] *Dipartimento di Informatica, Università degli Studi di Verona, Strada le Grazie 15, 37134 Verona, Italy*

Abstract

Cyber-Physical Systems (CPSs) are integrations of networking and distributed computing systems with physical processes, where feedback loops allow physical processes to affect computations and vice versa. Although CPSs can be found in several real-world domains (automotive, avionics, energy supply, etc), their verification often relies on *simulation test systems* rather then *formal methodologies*. This is because there is still a lack of research on the modelling and the definition of formal semantics to compare non-trivial CPSs in terms of their runtime behaviours up to an acceptable *tolerance*.

We propose a *hybrid probabilistic process calculus* for modelling and reasoning on CPSs. The dynamics of the calculus is expressed in terms of a *probabilistic labelled transition system* in the SOS style of Plotkin. This is used to define a *bisimulation-based* probabilistic behavioural semantics which supports compositional reasonings. For a more careful comparison between CPSs, we provide two compositional *probabilistic metrics* to formalise the notion of behavioural distance between systems, also in the case of bounded computations. Finally, we provide a non-trivial case study, taken from an engineering application, and use it to illustrate our definitions and our compositional behavioural theory for CPSs.

Keywords: Cyber-physical system, Hybrid probabilistic process calculus, Probabilistic metric semantics

1. Introduction

Cyber-Physical Systems (CPSs) are integrations of networking and distributed computing systems with physical processes, where feedback loops allow physical processes to affect computations and vice versa. CPSs can be considered as an evolution of *embedded systems*, where components are immersed in and interact with the physical world, via physical devices (such as *sensors* and *actuators*). They can also be seen as an evolution of *networked control systems*, where physical processes and controllers interact via a communication system.

The *physical plant* of a CPS is often represented by means of a *discrete-time state-space model*[1] consisting of two equations of the form

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + e_k \end{aligned}$$

where $x_k \in \mathbb{R}^n$ is the current *(physical) state*, $u_k \in \mathbb{R}^m$ is the *input* (i.e., the control actions implemented through actuators) and $y_k \in \mathbb{R}^p$ is the *output* (i.e., the measurements obtained from the sensors). The *uncertainty* $w_k \in \mathbb{R}^n$ and

[☆] A preliminary version appeared in the proceedings of the *11th International Conference on Language and Automata Theory and Applications* (LATA 2017), volume 10168 of *Lecture Notes in Computer Science*, pp. 115-127, Springer, 2017 [1].

[☆☆] Massimo Merro is partially supported by the project "Dipartimenti di Eccellenza 2018-2022", funded by the Italian Ministry of Education, Universities and Research (MIUR), and by the Joint Project 2017 "Security Static Analysis for Android Things", jointly funded by the University of Verona and JuliaSoft Srl.

[*] Corresponding author
Email address: massimo.merro@univr.it (Massimo Merro)

[1] We refer to [2] for a taxonomy of time-scale models used to represent CPSs.

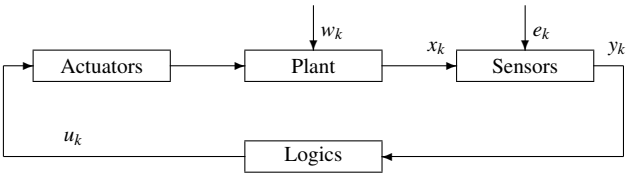

Figure 1: Structure of a CPS

the *measurement error* $e_k \in \mathbb{R}^p$ represent perturbation and sensor noise, respectively. The parameters A, B, and C are matrices modelling the dynamics of the physical system. The *next state* x_{k+1} depends on the current state x_k and the corresponding control actions u_k, at the sampling instant $k \in \mathbb{N}$. Note that, the state x_k cannot be directly observed: only its measurement y_k can be observed.

The physical plant is supported by a communication network through which the sensor measurements and actuator data are exchanged with the *controller(s)*, i.e., the *cyber* component, also called logics, of a CPS (see Figure 1).

In general terms, CPSs can be considered as both *nondeterministic* and *probabilistic* systems. Nondeterminism arises as they consist of distributed networks in which the activities of specific components occur nondeterministically, whereas the probabilistic behaviour is due to the presence of the uncertainty in the model and the measurement error, which are usually represented as *probability distributions*.

The range of CPSs applications is rapidly increasing and already covers several domains [3]: advanced automotive systems, energy conservation, environmental monitoring, avionics, critical infrastructure control (for instance, electric power, water resources, and communications systems), etc.

However, there is still a lack of research on the modelling and validation of CPSs through formal methodologies that allow us to model the interactions among the system components, and to verify the correctness of a CPS, as a whole, before its practical implementation. A straightforward utilisation of these methodologies is for *model-checking* [4], or even better, for *probabilistic model-checking* [5], to statically assess whether the current system deployment can guarantee the expected behaviour. However, they can also be an important aid for system planning, for instance to decide whether different deployments for a given application are behaviourally equivalent.

Process calculi have been successfully used to model and analyse concurrent, distributed and mobile systems (see, e.g., the *π-calculus* [6], *Ambients* [7] and the *Distributed π-calculus* [8]). However, to better describe systems based on a particular paradigm, dedicated calculi are needed. *Hybrid process algebras* [9, 10, 11, 12, 13] have been proposed for reasoning about physical systems and provide techniques for analysing and verifying protocols for hybrid automata. In order to enrich hybrid models with probabilistic or stochastic behaviour, a number of different approaches have been proposed in the last years [14, 15, 16, 17, 18, 19, 20]. However, to our knowledge, none of these formalisms provide bisimulation metrics semantics to estimate the deviation in terms of behaviour of different CPSs in a process-algebra setting. The definition of these instruments represents the main goal of the current paper.

Contributions. In this paper, we propose a *hybrid probabilistic process calculus*, called pCCPS (*Probabilistic Calculus of Cyber-Physical Systems*), with a clearly-defined *probabilistic behavioural semantics* for specifying and reasoning on CPSs. In pCCPS, cyber-physical systems are represented by making a neat distinction between the *physical component* describing the physical process (consisting in state variables, sensors, actuators, evolution law, measurement law, etc.) and the *cyber component*, i.e., the *logics* (i.e., controllers, IDS, supervisors, etc.) that governs sensor reading and actuator writing, as well as channel-based communication with other cyber components. Thus, channels are used for logical interactions between cyber components, whereas sensors and actuators make possible the interaction between cyber and physical components. Despite this conceptual similarity, messages transmitted via channels are "consumed" upon reception, whereas actuators' states (think of a valve) remains unchanged until the corresponding controller modifies them.

The calculus pCCPS adopts a *discrete notion of time* [21] and it is equipped with a *probabilistic labelled transition*

semantics (pLTS) in the style of [22]. We prove that our probabilistic labelled transition semantics satisfies some standard time properties such as: *time determinism*, *patience*, *maximal progress*, and *well-timedness*. Based on our pLTS, we define a natural notion of *weak probabilistic bisimilarity*, written $\approx$. As a main result, we prove that bisimilarity in pCCPS is preserved by appropriate system contexts and it is hence suitable for *compositional reasoning*.

Then, we provide a non-trivial *case study*, taken from an engineering application, and use it to illustrate our definitions and our compositional behavioural theory for CPSs. We also use our case study to show that the probabilistic bisimilarity is only partially satisfactory to reason on CPSs as it can only establish whether two CPSs behave exactly in the same way or not. Any tiny variation of the probabilistic behaviour of one of the two systems under consideration will break the equality without any further information on the "distance" of their behaviours. To this end, *bisimulation metric semantics* have been successfully employed to formalise the *behavioural distance* between two systems [23, 24, 25, 26].

We generalise our probabilistic bisimilarity by providing a notion of *weak bisimulation metric* for pCCPS along the lines of [23]. We will write $M \approx_p N$, if the weak bisimilarity between CPSs M and N holds with a *distance* p, with $p \in [0, 1]$. Intuitively, $\approx_0$ will coincide with the weak probabilistic bisimilarity $\approx$, whereas $\bigcup_{p\in[0,1]} \approx_p$ will correspond to the cartesian product pCCPS $\times$ pCCPS.

We also provide a notion of *n-bisimilarity metric* which takes into account bounded computations of systems [24]. This kind of metric, denoted with $\approx_p^n$, for $n \in \mathbb{N}^+$, says that the distance p of the systems under considerations is ensured only for the first n computation steps. Said in other words, if $M \approx_p^n N$ then for the first n computation steps the runtime behaviour of systems M and N may differ with *probability* at most p. Both metrics $\approx_p$ and $\approx_p^n$ are proved to be preserved by the same contexts considered for $\approx$, and hence they reveal to be suitable for compositional reasonings. In particular, they satisfy a well-known compositional property called *non-expansiveness* [24, 27, 28, 29], the analogue of the congruence property of weak bisimulation. Finally, with the help of our case study, we will show how n-bisimilarity metric can be very helpful in situations where it is not necessary to observe a system "ad infinitum" as it makes much more sense to observe its behaviour for bounded computations.

Outline. In Section 2, we give syntax and operational semantics of pCCPS. In Section 3, we provide a bisimulation-based probabilistic behavioural semantics for pCCPS and prove its compositionality. In Section 4, we model our case study in pCCPS, and prove for it run-time properties as well as system equalities. In Section 5, we define bisimulation metrics for pCCPS. In Section 6, we revise our case study by providing a more accurate analysis based on the proposed bisimulation metrics. In Section 7, we draw conclusions and discuss related and future work.

2. The calculus

In this section, we introduce our *Probabilistic Calculus of Cyber-Physical Systems*, pCCPS.

Let us start with some preliminary notations. We use $x, x_k \in \mathsf{X}$ for *state variables* (associated to physical states of systems), $c, d \in \mathsf{C}$ for *communication channels*, $a, a_k \in \mathsf{A}$ for *actuator devices*, $s, s_k \in \mathsf{S}$ for *sensors devices*. *Actuator names* are metavariables for actuator devices like *valve*, *light*, etc. Similarly, *sensor names* are metavariables for sensor devices, e.g., a sensor *thermometer* that measures a state variable called *temperature*, with a given precision. *Values*, ranged over by $v, v' \in \mathsf{V}$, are built from basic values, such as Booleans, integers and real numbers; they also include names. Given a generic set of names N, we write $\mathbb{R}^{\mathsf{N}}$ to denote the set of functions assigning a real value to each name in N. For $\xi \in \mathbb{R}^{\mathsf{N}}$, $n \in \mathsf{N}$ and $v \in \mathbb{R}$, we write $\xi[n \mapsto v]$ to denote the function $\psi \in \mathbb{R}^{\mathsf{N}}$ such that $\psi(m) = \xi(m)$, for any $m \neq n$, and $\psi(n) = v$. Given $\xi_1 \in \mathbb{R}^{\mathsf{N}_1}$ and $\xi_2 \in \mathbb{R}^{\mathsf{N}_2}$ such that $\mathsf{N}_1 \cap \mathsf{N}_2 = \emptyset$, we denote with $\xi_1 \uplus \xi_2$ the function in $\mathbb{R}^{\mathsf{N}_1 \cup \mathsf{N}_2}$ such that $(\xi_1 \uplus \xi_2)(n) = \xi_1(n)$, if $n \in \mathsf{N}_1$, and $(\xi_1 \uplus \xi_2)(n) = \xi_2(n)$, if $n \in \mathsf{N}_2$.

As pCCPS is a probabilistic calculus, we report the necessary mathematical machinery for its formal definition.

Definition 2.1 (Probability distribution)**.** *A (discrete)* probability sub-distribution *over a set of generic objects* O *is a function* $\delta \colon \mathsf{O} \to [0, 1]$ *with* $\sum_{o\in\mathsf{O}} \delta(o) \in (0, 1]$. *We write* $|\delta|$ *as an abbreviation for* $\sum_{o\in\mathsf{O}} \delta(o)$. *The* support *of a probability sub-distribution* δ *is given by* $\mathsf{supp}(\delta) = \{o \in \mathsf{O} \colon \delta(o) > 0\}$. *We write* $\mathcal{D}_{\text{sub}}(\mathsf{O})$*, ranged over* γ*,* δ *and* ϵ*, for the set of all* finite-support *probability sub-distributions over the set* O. *A probability sub-distribution* $\delta \in \mathcal{D}_{\text{sub}}(\mathsf{O})$ *is said to be a* probability distribution *if* $\sum_{o\in\mathsf{O}} \delta(o) = 1$. *With* $\mathcal{D}(\mathsf{O})$ *we denote the set of all finite-support probability distributions over* O. *For any* $o \in \mathsf{O}$*, the* point (Dirac) distribution at o*, denoted* $\overline{o}$*, assigns probability* 1 *to* o *and* 0 *to all other elements of* O*, so that* $\mathsf{supp}(\overline{o}) = \{o\}$.

Let I be a finite indexing set such that: (i) δ_i is a sub-distribution in $\mathcal{D}_{\text{sub}}(\mathsf{O})$ for each $i \in I$, and (ii) $p_i \geq 0$ are probabilities such that $\sum_{i\in I} p_i \in (0, 1]$. The probability sub-distribution (or convex combination) $\sum_{i\in I} p_i \cdot \delta_i$ is the sub-distribution defined by $(\sum_{i\in I} p_i \cdot \delta_i)(o) = \sum_{i\in I} p_i \delta_i(o)$ for all $o \in \mathsf{O}$. We write a sub-distribution as $p_1 \cdot \delta_1 + \ldots + p_k \cdot \delta_k$ when the indexing set I is $\{1, \ldots, k\}$.

In pCCPS, a cyber-physical system consists of:

- a *physical component* (defining physical variables, physical devices, physical evolution, etc.) and
- a *cyber (or logical) component* that interacts with the physical devices (sensors and actuators) and communicates via channels with other cyber components.

Physical components in pCCPS are given by two sub-components: (i) the *physical state*, which is supposed to change at runtime, and (ii) the *physical environment*, which contains static information[2].

Definition 2.2 (Physical state)**.** *Let* X *be a set of state variables,* S *be a set of sensors, and* A *be a set of actuators. A* physical state S *is a triple* $\langle \xi_{\text{x}}, \xi_{\text{s}}, \xi_{\text{a}} \rangle$*, where:*

- $\xi_{\text{x}} \in \mathbb{R}^{\mathsf{X}}$ *is the* state function*,*
- $\xi_{\text{s}} \in \mathbb{R}^{\mathsf{S}}$ *is the* sensor function*,*
- $\xi_{\text{a}} \in \mathbb{R}^{\mathsf{A}}$ *is the* actuator function*.*

All functions defining a physical state are total*.*

The *state function* ξ_{x} returns the current value associated to each variable in X. The *sensor function* ξ_{s} returns the current value associated to each sensor in S; similarly, the *actuator function* ξ_{a} returns the current value associated to each actuator in A.

Definition 2.3 (Physical environment)**.** *Let* X *be a set of state variables,* S *be a set of sensors, and* A *be a set of actuators. A* physical environment E *is a triple* $\langle evol, meas, inv \rangle$*, where:*

- $evol\colon \mathbb{R}^{\mathsf{X}} \times \mathbb{R}^{\mathsf{A}} \to \mathcal{D}(\mathbb{R}^{\mathsf{X}})$ *is the* evolution map*,*
- $meas\colon \mathbb{R}^{\mathsf{X}} \to \mathcal{D}(\mathbb{R}^{\mathsf{S}})$ *is the* measurement map*,*
- $inv \in 2^{\mathbb{R}^{\mathsf{X}}}$ *is the* invariant set*.*

All the functions defining a physical environment are total functions*.*

Given a state function and an actuator function, the *evolution map evol* returns a *probability distribution over state functions*. This function models the *evolution law* of the physical system, where changes made on actuators may reflect on state variables. Since we assume the presence of a known (maximal) uncertainty for our models, the evolution map does not return a specific state function but a probability distribution over state functions.

Given a state function, the *measurement map meas* returns a *probability distribution over sensor functions*. Also in this case, since we assume the presence of a known (maximal) measurement error for each sensor, the measurement map returns a probability distribution over sensor functions, rather than a specific sensor function.

The *invariant set inv* returns the set of state functions that satisfy the invariant of the system. A CPS that gets into a physical state with a state function that does not satisfy the invariant is in *deadlock*.

Let us now formalise the cyber components of CPSs in our calculus pCCPS. Our (logical) processes build on Hennessy and Regan's *Timed Process Language* TPL [21] (basically CCS enriched with a discrete notion of time). We extend TPL with three constructs: one to read values detected at sensors, one to write values on actuators, and one to express (guarded) probabilistic choice. The remaining processes of the calculus are the same as those of TPL.

[2] Actually, this information is periodically updated (say, every six months) to take into account possible drifts of the system.

Definition 2.4 (Processes)**.** Processes *are defined by the grammar:*

$$\begin{array}{rcl} P, Q & ::= & \mathsf{nil} \mid \mathsf{tick}.C \mid P \parallel Q \mid \lfloor chn.C \rfloor D \mid phy.C \mid [b]\{P\},\{Q\} \mid P\backslash c \mid X \mid \mathsf{rec}\, X.P \\ C, D & ::= & \bigoplus_{i\in I} p_i{:}P_i \\ chn & ::= & \mathsf{snd}\, c\langle v\rangle \mid \mathsf{rcv}\, c(x) \\ phy & ::= & \mathsf{read}\, s(x) \mid \mathsf{write}\, a\langle v\rangle\,. \end{array}$$

We write nil for the *terminated process*. The process $\mathsf{tick}.C$ models sleeping for one time unit. We write $P \parallel Q$ to denote the *parallel composition* of concurrent processes P and Q. The process $\lfloor chn.C \rfloor D$, with $chn \in \{\mathsf{snd}\, c\langle v\rangle, \mathsf{rcv}\, c(x)\}$, denotes *channel transmission with timeout*. Thus, $\lfloor \mathsf{snd}\, c\langle v\rangle.C \rfloor D$ sends the value v on channel c and, after that, it continues as C; otherwise, if no communication partner is available within one time unit, it evolves into D. The process $\lfloor \mathsf{rcv}\, c(x).C \rfloor D$ is the obvious counterpart for channel reception.

Processes of the form $phy.C$ denote activities on physical devices (sensors or actuators). Thus, the construct $\mathsf{read}\, s(x).C$ reads the value v detected by the sensor s and, after that, it continues as C, where x is replaced by v. The process $\mathsf{write}\, a\langle v\rangle.C$ writes the value v on the actuator a and then it continues as C.

The process $P\backslash c$ is the channel restriction operator of CCS. It is quantified over the set of communication channels, although we often use the shorthand $P\backslash\{c_1, \cdots, c_n\}$ to mean $P\backslash c_1\backslash c_2 \cdots \backslash c_n$. The process $[b]\{P\},\{Q\}$ is the standard conditional, where b is a decidable guard. For simplicity, as in CCS, we identify process $[b]\{P\},\{Q\}$ with P, if b evaluates to true, and $[b]\{P\},\{Q\}$ with Q, if b evaluates to false. In processes of the form $\mathsf{tick}.D$ and $\lfloor chn.C \rfloor D$, the occurrence of D is said to be *time-guarded*. The process $\mathsf{rec}\, X.P$ denotes *time-guarded recursion* as all occurrences of the process variable X may only occur time-guarded in P.

The construct $\bigoplus_{i\in I} p_i{:}P_i$ denotes *probabilistic choice*, where I is a *finite*, *non-empty* set of indexes, and $p_i \in (0, 1]$, for $i \in I$, denotes the probability to execute the process P_i, with $\sum_{i\in I} p_i = 1$. As in [30], in order to simplify the operational semantics, *probabilistic choices occur always underneath prefixing*.

In the two constructs $\lfloor \mathsf{rcv}\, c(x).C \rfloor D$ and $\mathsf{read}\, s(x).C$, the variable x is said to be *bound*. Similarly, the process variable X is bound in $\mathsf{rec}\, X.P$. This gives rise to the standard notions of *free/bound (process) variables* and *α-conversion*. We identify processes up to α-conversion (similarly, we identify CPSs up to renaming of state variables, sensor names, and actuator names). A term is *closed* if it does not contain free (process) variables, and we assume to always work with closed processes: the absence of free variables is preserved at run-time. As further notation, we write $T\{^{v}/_{x}\}$ for the substitution of the variable x with the value v in any expression T of our language. Similarly, $T\{^{P}/_{X}\}$ is the substitution of the process variable X with the process P in T.

Everything is in place to provide the definition of cyber-physical systems expressed in pCCPS.

Definition 2.5 (Cyber-physical system)**.** *Fixed a set of state variables* X*, a set of sensors* S*, and a set of actuators* A*, a* cyber-physical system *in pCCPS is given by two components:*

- *a* physical component *consisting of*
  - *a* physical environment *E defined on* X*,* S*, and* A*, and*
  - *a* physical state *S recording the current values associated to the state variables in* X*, the sensors in* S*, and the actuators in* A*;*
- *a* cyber component *P that interacts with the sensors in* S *and the actuators* A*, and can communicate, via channels, with other cyber components of the same or of other CPSs.*

We write $E; S \bowtie P$ to denote the resulting CPS, and use M and N to range over CPSs. Sometimes, when the physical environment E is clearly identified, we write $S \bowtie P$ instead of $E; S \bowtie P$. CPSs of the form $S \bowtie P$ are called environment-free CPSs.

The reader should notice that the syntax of our CPSs is slightly too permissive as a process might use sensors and/or actuators which are not defined in the physical state.

Definition 2.6 (Well-formedness)**.** *Let $S = \langle \xi_{\mathrm{x}}, \xi_{\mathrm{s}}, \xi_{\mathrm{a}} \rangle$ be a physical state, $E = \langle evol, meas, inv \rangle$ a physical environment, and P a process. The CPS $E; S \bowtie P$ is said to be* well-formed *if: (i) any sensor mentioned in P is in the domain of the function ξ_{s}; (ii) any actuator mentioned in P is in the domain of the function ξ_{a}. A sub-distribution $\gamma \in \mathcal{D}_{\mathrm{sub}}(\mathtt{pCCPS})$ is said to be well-formed if its support contains only well-formed CPSs.*

$$\text{(Outp)}\ \frac{-}{\lfloor \mathsf{snd}\, c\langle v\rangle.C \rfloor D \xrightarrow{\overline{c}v} [\![C]\!]} \qquad \text{(Inpp)}\ \frac{-}{\lfloor \mathsf{rcv}\, c(x).C \rfloor D \xrightarrow{cv} [\![C\{^{v}/_{x}\}]\!]}$$

$$\text{(Write)}\ \frac{-}{\mathsf{write}\, a\langle v\rangle.C \xrightarrow{a!v} [\![C]\!]} \qquad \text{(Read)}\ \frac{-}{\mathsf{read}\, s(x).C \xrightarrow{s?(x)} [\![C]\!]}$$

$$\text{(Com)}\ \frac{P_1 \xrightarrow{\overline{c}v} \pi_1 \quad P_2 \xrightarrow{cv} \pi_2}{P_1 \parallel P_2 \xrightarrow{\tau} \pi_1 \parallel \pi_2} \qquad \text{(Par)}\ \frac{P \xrightarrow{\lambda} \pi \quad \lambda \neq \mathsf{tick}}{P \parallel Q \xrightarrow{\lambda} \pi \parallel \overline{Q}}$$

$$\text{(ChnRes)}\ \frac{P \xrightarrow{\lambda} \pi \quad \lambda \notin \{cv, \overline{c}v\}}{P\backslash c \xrightarrow{\lambda} \pi\backslash c} \qquad \text{(Rec)}\ \frac{P\{^{\mathsf{rec}\, X.P}/_{X}\} \xrightarrow{\lambda} \pi}{\mathsf{rec}\, X.P \xrightarrow{\lambda} \pi}$$

$$\text{(TimeNil)}\ \frac{-}{\mathsf{nil} \xrightarrow{\mathsf{tick}} \overline{\mathsf{nil}}} \qquad \text{(Delay)}\ \frac{-}{\mathsf{tick}.C \xrightarrow{\mathsf{tick}} [\![C]\!]}$$

$$\text{(Timeout)}\ \frac{-}{\lfloor chn.C \rfloor D \xrightarrow{\mathsf{tick}} [\![D]\!]} \qquad \text{(TimePar)}\ \frac{P_1 \xrightarrow{\mathsf{tick}} \pi_1 \quad P_2 \xrightarrow{\mathsf{tick}} \pi_2 \quad P_1 \parallel P_2 \xrightarrow{\tau} \not\to}{P_1 \parallel P_2 \xrightarrow{\mathsf{tick}} \pi_1 \parallel \pi_2}$$

Table 1: Probabilistic LTS for processes

Hereafter, we will always work with well-formed CPSs.

As usual in process calculi, we use the symbol $\equiv$ to denote standard *structural congruence* for timed processes [6, 31]; its generalisation to CPSs is immediate: $E; S \bowtie P \equiv E; S \bowtie Q$ if $P \equiv Q$. Also the generalisation to sub-distributions in $\mathcal{D}_{\text{sub}}(\mathtt{pCCPS})$ is straightforward: given two sub-distributions γ and γ' over CPSs, we write $\gamma \equiv \gamma'$ if $\gamma([M]_{\equiv}) = \gamma'([M]_{\equiv})$ for all equivalence classes $[M]_{\equiv} \subseteq \mathtt{pCCPS}$.

Finally, we assume a number of *notational conventions*. We write Dead to denote a deadlocked CPS which cannot perform any action. This fictitious CPS will be useful when defining behavioural distances between CPSs (see Definition 5.3). We write $chn.P$ instead of $\mathsf{rec}\, X.\lfloor chn.P \rfloor X$, when X does not occur in P. We write $\mathsf{snd}\, c$ (resp. $\mathsf{rcv}\, c$) when channel c is used for pure synchronisation. For $k \geq 0$, we write $\mathsf{tick}^k.P$ as a shorthand for $\mathsf{tick}.\mathsf{tick}.\ldots.\mathsf{tick}.P$, where the prefix tick appears k consecutive times. Given a CPS $M = E; S \bowtie P$, a process Q and a channel c, we write $M \parallel Q$ for $E; S \bowtie (P \parallel Q)$, and $M\backslash c$ for $E; S \bowtie (P\backslash c)$.

In the rest of the paper, symbol σ ranges over distributions over physical states, π ranges over distributions over processes, and γ ranges over distributions over CPSs.

2.1. Probabilistic labelled transition semantics

In this section, we provide the dynamics of pCCPS in terms of a *probabilistic labelled transition system* (pLTS) [22]. First, we give a pretty standard probabilistic LTS for processes, then we lift transition rules from processes to CPSs to deal with the probability distributions occurring in physical environments.

In Table 1, we provide transition rules for processes. Here, the meta-variable λ ranges over labels in the set $\{\mathsf{tick}, \tau, \overline{c}v, cv, a!v, s?(x)\}$. These labels denote the passage of time, internal activities, channel transmission, channel reception, actuator writing, and sensor reading, respectively. As in [30], the definition of the labelled transition relation for processes relies on a semantic interpretation of probabilistic processes in terms of (discrete) probability distributions over processes.

Definition 2.7. *For any probabilistic choice $\bigoplus_{i\in I} p_i{:}P_i$ over a finite index set I, we write $[\![\bigoplus_{i\in I} p_i{:}P_i]\!]$ to denote the probability distribution $\sum_{i\in I} p_i \cdot \overline{P_i}$.*

The transition rules in Table 1 use some obvious notation for distributing both parallel composition and channel restriction over a sub-distribution. Given two sub-distributions π_1 and π_2 we define the sub-distribution $\pi_1 \parallel \pi_2$ as follows: $(\pi_1 \parallel \pi_2)(P) = \pi_1(P_1) \cdot \pi_2(P_2)$, if $P = P_1 \parallel P_2$; $(\pi_1 \parallel \pi_2)(P) = 0$, otherwise. Given an arbitrary distribution

$$\text{(Out)}\ \frac{P \xrightarrow{\overline{c}v} \pi \qquad S \in inv}{S \bowtie P \xrightarrow{\overline{c}v} \overline{S} \bowtie \pi} \qquad \text{(Inp)}\ \frac{P \xrightarrow{cv} \pi \qquad S \in inv}{S \bowtie P \xrightarrow{cv} \overline{S} \bowtie \pi} \qquad \text{(Tau)}\ \frac{P \xrightarrow{\tau} \pi \qquad S \in inv}{S \bowtie P \xrightarrow{\tau} \overline{S} \bowtie \pi}$$

$$\text{(SensRead)}\ \frac{P \xrightarrow{s?(z)} \pi \qquad \xi_{\mathrm{s}}(s) = \sum_{i\in I} p_i \cdot \overline{v_i} \qquad \xi_{\mathrm{x}} \in inv}{\langle \xi_{\mathrm{x}}, \xi_{\mathrm{s}}, \xi_{\mathrm{a}} \rangle \bowtie P \xrightarrow{\tau} \overline{\langle \xi_{\mathrm{x}}, \xi_{\mathrm{s}}, \xi_{\mathrm{a}} \rangle} \bowtie \sum_{i\in I} p_i \cdot \pi\{^{v_i}/_z\}}$$

$$\text{(ActWrite)}\ \frac{P \xrightarrow{a!v} \pi \qquad \xi_{\mathrm{x}} \in inv}{\langle \xi_{\mathrm{x}}, \xi_{\mathrm{s}}, \xi_{\mathrm{a}} \rangle \bowtie P \xrightarrow{\tau} \overline{\langle \xi_{\mathrm{x}}, \xi_{\mathrm{s}}, \xi_{\mathrm{a}}[a \mapsto v] \rangle} \bowtie \pi}$$

$$\text{(Time)}\ \frac{P \xrightarrow{\mathsf{tick}} \pi \qquad S \bowtie P \not\xrightarrow{\tau} \qquad S \in inv}{S \bowtie P \xrightarrow{\mathsf{tick}} next_E(S) \bowtie \pi} \qquad \text{(Deadlock)}\ \frac{S \notin inv}{S \bowtie P \xrightarrow{\tau} \overline{\mathsf{Dead}}}$$

Table 2: Probabilistic LTS for a CPS $S \bowtie P$ parametric on an environment $E = \langle evol, meas, inv \rangle$

over processes $\pi = \sum_{i\in I} p_i \cdot \overline{P_i}$, an arbitrary channel c, and a value v, we define $\pi\backslash c$ as the distribution $\sum_{i\in I} p_i \cdot \overline{P_i\backslash c}$, and $\pi\{^v/_x\}$ as the distribution $\sum_{i\in I} p_i \cdot \overline{P_i\{^v/_x\}}$.

Let us comment on the transition rules of Table 1. Rules (Outp), (Inpp) and (Com) serve to model channel communication, on some channel c. Rule (Write) denotes the writing of some data v on an actuator a. Rule (Read) denotes the reading of some value detected at sensor s. Rule (Par) propagates untimed actions over parallel components. Rules (ChnRes) and (Rec) are the standard rules for channel restriction and recursion, respectively. The following four rules are standard, and model the passage of one time unit. The symmetric counterparts of rules (Com) and (Par) are obvious and thus omitted from the table.

In Table 2, we lift the transition rules from processes to systems, actually to probability distributions over systems. We adopt the following notation for probability distributions: given a distribution σ over physical states and a distribution π over processes, we write $\sigma \bowtie \pi$ to denote the distribution over (environment-free) CPSs defined as $(\sigma \bowtie \pi)(S \bowtie P) = \sigma(S) \cdot \pi(P)$. Moreover, given a physical environment E, we write $E; \sigma \bowtie \pi$ to extend the distribution $\sigma \bowtie \pi$ to full CPSs as follows: $(E; \sigma \bowtie \pi)(E; S \bowtie P) = \sigma(S) \cdot \pi(P)$. Actions, ranged over by α, are in the set $Act = \{\tau, \overline{c}v, cv, \mathsf{tick}\}$. These actions denote: non-observable activities (τ); channel transmission ($\overline{c}v$); channel reception (cv); the passage of time (tick).

As physical environments contain static information, for simplicity the resulting transition rules are parameterised on a physical environment of the form $E = \langle evol, meas, inv \rangle$. Thus, instead of providing the transitions rules for a CPS of the form $E; S \bowtie P$ we give the LTS semantics parametric on E for the environment-free CPS $S \bowtie P$.

All rules, except (Deadlock), have a common premise requiring that the current state function of the system must satisfy the invariant. With an abuse of notation, we sometimes write $S \in inv$ instead of $\xi_{\mathrm{x}} \in inv$ when $S = \langle \xi_{\mathrm{x}}, \xi_{\mathrm{s}}, \xi_{\mathrm{a}} \rangle$. Rules (Out) and (Inp) model transmission and reception, with an external system, on a channel c. Rule (Tau) lifts non-observable actions from processes to systems. Rule (SensRead) models the reading of the current data detected at sensor s. Rule (ActWrite) models the writing of a value v on an actuator a. A similar lifting occurs in rule (Time) for timed actions, where $next_E(S)$ returns a probability distribution over possible physical states for the next time slot, according to the current physical state S and physical environment E. Formally, for $S = \langle \xi_{\mathrm{x}}, \xi_{\mathrm{s}}, \xi_{\mathrm{a}} \rangle$ and $E = \langle evol, meas, inv \rangle$, we define:

$$next_E(S) \;=\; \sum_{\substack{\xi'_{\mathrm{x}} \in \mathsf{supp}(evol(\xi_{\mathrm{x}}, \xi_{\mathrm{a}})) \\ \xi'_{\mathrm{s}} \in \mathsf{supp}(meas(\xi'_{\mathrm{x}}))}} (evol(\xi_{\mathrm{x}}, \xi_{\mathrm{a}})(\xi'_{\mathrm{x}}) \cdot meas(\xi'_{\mathrm{x}})(\xi'_{\mathrm{s}})) \cdot \overline{\langle \xi'_{\mathrm{x}}, \xi'_{\mathrm{s}}, \xi_{\mathrm{a}} \rangle}\,.$$

Intuitively, the operator $next_E$ serves to compute the possible state functions and sensor functions of the next time slot (actuator changes are governed by the cyber-component). More precisely, the (probability distribution over the) next state function is determined by applying $evol$ to the current state function ξ_{x} and the current actuator function ξ_{a}. The probability weight of any possible state function ξ'_{x} is given by $evol(\xi_{\mathrm{x}}, \xi_{\mathrm{a}})(\xi'_{\mathrm{x}})$. Then, for a state function ξ'_{x}, the (probability distribution over the) next sensor function is given by applying $meas$ to ξ'_{x}. Finally, the probability weight of any possible sensor function ξ'_{s} is given by $meas(\xi'_{\mathrm{x}})(\xi'_{\mathrm{s}})$.

Recapitulating, by an application of rule (Time) a CPS moves to the next physical state, in the next time slot. Rule (Deadlock) is straightforward: if the invariant is not satisfied then the CPS deadlocks.

Finally, notice that in our LTS we defined transitions rules of the form $S \bowtie P \xrightarrow{\alpha} \sigma \bowtie \pi$, parametric on some physical environment E. As physical environments do not change at runtime, $S \bowtie P \xrightarrow{\alpha} \sigma \bowtie \pi$ entails $E; S \bowtie P \xrightarrow{\alpha} E; \sigma \bowtie \pi$, thus providing the probabilistic LTS for (full) CPSs.

Remark 2.8. *Note that the rules in Table 2 define an* image-finite *pLTS. This means that for any CPS M and label α there are finitely many distributions reachable from M in one α-labelled transition step. Moreover, all transitions $M \xrightarrow{\alpha} \gamma$ are such that γ has a finite support.*

Now, having defined the labelled transitions that can be performed by a CPS of the form $E; S \bowtie P$, we can easily concatenate these transitions to define the possible computation traces of a system. A *computation trace* [32] for a CPS $E; S_1 \bowtie P_1$ is a sequence of steps of the form $E; S_1 \bowtie P_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} E; S_n \bowtie P_n$ where for any i, with $1 \leq i \leq n-1$, we have $E; S_i \bowtie P_i \xrightarrow{\alpha_i} E; \sigma_{i+1} \bowtie \pi_{i+1}$ for distributions σ_{i+1} and π_{i+1} such that $S_{i+1} \in \mathsf{supp}(\sigma_{i+1})$ and $P_{i+1} \in \mathsf{supp}(\pi_{i+1})$.

Below, we report a few desirable time properties [21] which hold in our calculus: (a) *time determinism*, (b) *maximal progress*, (c) *patience*, and (d) *well-timedness*. In its standard formulation, *time determinism* says that a system reaches at most one new state by executing a timed action tick; however, in our setting, this holds only for the logical components (up to structural congruence) whereas the evolution of the physical component is intrinsically probabilistic, due to the presence of uncertainty and measurement errors. The *maximal progress* property usually says that processes communicate as soon as a possibility of communication arises. In our calculus, we generalise this property saying that instantaneous (silent) actions cannot be delayed. On the other hand, *patience* says that if no instantaneous actions are possible then time is free to pass. Finally, *well-timedness* [31, 33] ensures the absence of infinite instantaneous traces which would prevent the passage of time, and hence the physical evolution of a CPS.

Theorem 2.9 (Time properties)**.** *Let $M = E; S \bowtie P$.*

(a) If $M \xrightarrow{\mathsf{tick}} \gamma$ and $M \xrightarrow{\mathsf{tick}} \gamma'$ then $\gamma \equiv \gamma'$.
(b) If $M \xrightarrow{\tau} \gamma$ then there is no γ' such that $M \xrightarrow{\mathsf{tick}} \gamma'$.
(c) If $M \xrightarrow{\mathsf{tick}} \gamma$ for no γ then either S does not satisfy the invariant of E or there is γ' such that $M \xrightarrow{\tau} \gamma'$.
(d) There is a $k \in \mathbb{N}$ such that if $M \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} N$, with $\alpha_i \neq \mathsf{tick}$, then $n \leq k$.

The proof of Theorem 2.9 can be found in the Appendix, in Section Appendix A.1.

3. Probabilistic bisimulation

In this section, we are ready to define a bisimulation-based behavioural equality for CPSs, relying on our labelled transition semantics. We recall that the only *observable activities* in pCCPS are: the passage of time and channel communication. As a consequence, the capability to observe physical events (different from deadlocks) depends on the capability of the cyber components to recognise those events by acting on sensors and actuators, and then signalling them using (unrestricted) channels.

In a probabilistic setting, the definition of weak transition $\stackrel{\hat{\alpha}}{\Longrightarrow}$, which abstract away non-observable actions, is complicated by the fact that (strong) transitions take CPSs to distributions over CPSs. Following [30, 34, 35], we need to generalise transitions, so that they take sub-distributions to sub-distributions.

With an abuse of notation, we use γ and γ' to range over sub-distributions over CPSs, under the assumption that $\sum_{M \in \mathtt{pCCPS}} \gamma(M) \leq 1$.

Let us start by defining the weak transition $M \xrightarrow{\hat{\alpha}} \gamma$ for any CPS M and distribution γ. If $\alpha = \tau$ then we write $M \xrightarrow{\hat{\alpha}} \gamma$ whenever either $M \xrightarrow{\alpha} \gamma$ or $\gamma = \overline{M}$. Otherwise, if $\alpha \neq \tau$ then we write $M \xrightarrow{\hat{\alpha}} \gamma$ whenever $M \xrightarrow{\alpha} \gamma$. The relation $\xrightarrow{\hat{\alpha}}$ is extended to model transitions from sub-distributions to sub-distributions. For a sub-distribution $\gamma = \sum_{i \in I} p_i \cdot \overline{M_i}$, we write $\gamma \xrightarrow{\hat{\alpha}} \gamma'$ if there is a non-empty set $J \subseteq I$ such that $M_j \xrightarrow{\hat{\alpha}} \gamma_j$ for all $j \in J$, $M_i \not\xrightarrow{\hat{\alpha}}$, for all $i \in I \setminus J$, and

$\gamma' = \sum_{j\in J} p_j \cdot \gamma_j$. Note that if $\alpha \neq \tau$ then this definition entails that only some CPSs in the support of γ have an $\xrightarrow{\hat{\alpha}}$ transition. Then, we define the weak transition relation $\stackrel{\hat{\tau}}{\Longrightarrow}$ as the transitive and reflexive closure of $\xrightarrow{\hat{\tau}}$, i.e. $\stackrel{\hat{\tau}}{\Longrightarrow} = (\xrightarrow{\hat{\tau}})^*$, while for $\alpha \neq \tau$ we let $\stackrel{\hat{\alpha}}{\Longrightarrow}$ denote $\stackrel{\hat{\tau}}{\Longrightarrow}\xrightarrow{\hat{\alpha}}\stackrel{\hat{\tau}}{\Longrightarrow}$.

In order to define a probabilistic bisimulation, following [36] we rely on the notion of *matching* [37] (also known as *coupling*) for a pair of distributions. Intuitively, the matching for a pair (γ, γ') may be understood as a transportation schedule for the shipment of probability mass from γ to γ'.

Definition 3.1 (Matching)**.** *A* matching *for a pair of distributions (γ, γ'), with $\gamma, \gamma' \in \mathcal{D}(\mathtt{pCCPS})$, is a distribution ω in the product space $\mathcal{D}(\mathtt{pCCPS} \times \mathtt{pCCPS})$ such that:*

- $\sum_{M' \in \mathtt{pCCPS}} \omega(M, M') = \gamma(M)$*, for all* $M \in \mathtt{pCCPS}$*, and*
- $\sum_{M \in \mathtt{pCCPS}} \omega(M, M') = \gamma'(M')$*, for all* $M' \in \mathtt{pCCPS}$*.*

We write $\Omega(\gamma, \gamma')$ to denote the set of all matchings for (γ, γ').

Everything is in place to define weak probabilistic bisimulation for pCCPS, along the lines of [38].

Definition 3.2 (Weak probabilistic bisimulation)**.** *A binary symmetric relation $\mathcal{R}$ over CPSs is a* weak probabilistic bisimulation *if $M \mathcal{R} N$ and $M \xrightarrow{\alpha} \gamma$ implies that there exist a distribution γ' and a matching $\omega \in \Omega(\gamma, \gamma')$ such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} \gamma'$, and $M' \mathcal{R} N'$ whenever $\omega(M', N') > 0$. We say that M and N are bisimilar, written $M \approx N$, if $M \mathcal{R} N$ for some weak probabilistic bisimulation $\mathcal{R}$.*

A main result of the paper is that bisimilarity can be used to reason on CPSs in a compositional manner. In particular, bisimilarity is preserved by parallel composition of *physically-disjoint* CPSs, by parallel composition of *pure-logical* processes, and by channel restriction; basically, all those contexts that cannot interfere on physical devices (sensors and actuators), whereas interferences on logical components (via channel communication) is allowed.

Intuitively, two CPSs are physically-disjoint if they have different plants but they may share logical channels for communication purposes. More precisely, physically-disjoint CPSs have disjoint state variables and disjoint physical devices (sensors and actuators). As we consider only well-formed CPSs (Definition 2.6), this ensures us that a CPS cannot physically interfere with a parallel CPS by acting on its physical devices. Although, logical interferences on communication channels are allowed.

Formally, let $S^i = \langle \xi^i_{\mathrm{x}}, \xi^i_{\mathrm{s}}, \xi^i_{\mathrm{a}} \rangle$ and $E^i = \langle evol^i, meas^i, inv^i \rangle$ be physical states and physical environments, respectively, associated to state variables in the set X_i, sensors in the set S_i, and actuators in the set A_i, for $i \in \{1, 2\}$. For $\mathsf{X}_1 \cap \mathsf{X}_2 = \emptyset$, $\mathsf{S}_1 \cap \mathsf{S}_2 = \emptyset$ and $\mathsf{A}_1 \cap \mathsf{A}_2 = \emptyset$, we define:

- the *disjoint union* of the physical states S_1 and S_2, written $S_1 \uplus S_2$, to be the physical state $\langle \xi_{\mathrm{x}}, \xi_{\mathrm{s}}, \xi_{\mathrm{a}} \rangle$ such that: $\xi_{\mathrm{x}} = \xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}}$, $\xi_{\mathrm{s}} = \xi^1_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}}$, and $\xi_{\mathrm{a}} = \xi^1_{\mathrm{a}} \uplus \xi^2_{\mathrm{a}}$;
- the *disjoint union* of the physical environments E_1 and E_2, written $E_1 \uplus E_2$, to be the physical environment $\langle evol, meas, inv \rangle$ such that:

$$\begin{array}{rcl} (evol(\xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}}, \xi^1_{\mathrm{a}} \uplus \xi^2_{\mathrm{a}}))(\xi^{1\prime}_{\mathrm{x}} \uplus \xi^{2\prime}_{\mathrm{x}}) & = & evol^1(\xi^1_{\mathrm{x}}, \xi^1_{\mathrm{a}})(\xi^{1\prime}_{\mathrm{x}}) \cdot evol^2(\xi^2_{\mathrm{x}}, \xi^2_{\mathrm{a}})(\xi^{2\prime}_{\mathrm{x}}) \\ (meas(\xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}}))(\xi^{1\prime}_{\mathrm{s}} \uplus \xi^{2\prime}_{\mathrm{s}}) & = & meas^1(\xi^1_{\mathrm{x}})(\xi^{1\prime}_{\mathrm{s}}) \cdot meas^2(\xi^2_{\mathrm{x}})(\xi^{2\prime}_{\mathrm{s}}) \\ \xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}} \in inv & \text{iff} & \xi^1_{\mathrm{x}} \in inv^1 \text{ and } \xi^2_{\mathrm{x}} \in inv^2 \,. \end{array}$$

Definition 3.3 (Physically-disjoint CPSs)**.** *Let $M_i = E_i; S_i \bowtie P_i$, for $i \in \{1, 2\}$. We say that M_1 and M_2 are physically-disjoint if S_1 and S_2 have disjoint sets of state variables, sensors and actuators. In this case, we write $M_1 \uplus M_2$ to denote the CPS defined as $(E_1 \uplus E_2); (S_1 \uplus S_2) \bowtie (P_1 \parallel P_2)$. For any $M \in \mathtt{pCCPS}$, the special system* Dead *is physically-disjoint with M, and $M \uplus \mathsf{Dead} = \mathsf{Dead} \uplus M = \mathsf{Dead}$.*

A *pure-logical process* is a process which may interfere on communication channels but it never interferes on physical devices as it never accesses sensors and/or actuators. Basically, a pure-logical process is a (possibly probabilistic) TPL process [21]. Thus, in a system $M \parallel Q$, where M is an arbitrary CPS, a pure-logical process Q cannot interfere with the physical evolution of M. Although, process Q can definitely interact with M via communication channels, and hence affect its observable behaviour.

Definition 3.4 (Pure-logical processes)**.** *A process P is called* pure-logical *if it never acts on sensors and/or actuators.*

Now, we can finally prove the compositionality of probabilistic bisimilarity $\approx$.

Theorem 3.5 (Congruence results)**.** *Let M and N be two arbitrary CPSs in* pCCPS*.*

1. *$M \approx N$ implies $M \uplus O \approx N \uplus O$, for any physically-disjoint CPS O;*
2. *$M \approx N$ implies $M \parallel P \approx N \parallel P$, for any pure-logical process P;*
3. *$M \approx N$ implies $M\backslash c \;\approx\; N\backslash c$, for any channel c.*

The proof can be found in the Appendix, at the end of Section Appendix A.3.

The reader may wonder whether the bisimilarity $\approx$ is preserved by more permissive contexts. The answer is no. Suppose to allow in the second item of Theorem 3.5 a process P that can also read on sensors. In this case, even if M and N are bisimilar, the parallel process P might read a different value in the two systems at the very same sensor s (due to the sensor error) and transmit these different values on a free channel, breaking the congruence. Activities on actuators may also lead to different behaviours of the compound systems: bisimilar CPSs may have physical components that are not exactly aligned. A similar reasoning applies when composing CPSs with non physically-disjoint ones: interference on physical devices may break the congruence.

However, in the next section we will see that the congruence results of Theorem 3.5 will be very useful when reasoning on complex systems.

4. Case study

In this section, we provide a case study to illustrate how pCCPS can be used to specify and reason on CPSs in a compositional manner. In particular, we model an engine whose temperature is maintained within a specific range by means of a cooling system.

As regards the *physical environment* we adopt discrete uniform distributions over suitable intervals to model both the evolution map and the measurement map[3]. In our model, we assume a *granularity* $g \in \mathbb{N}^+$ representing the precision 10^{-g} of the model in estimating physical values. Thus, for an arbitrary real interval $[v, w]$ we write $[v, w]_g$ to denote the *finite* set of reals $\{k \in [v, w] \colon k = v + h \cdot 10^{-g},\ \text{with } h \in \mathbb{N}\}$.

Given a granularity $g \in \mathbb{N}^+$, the physical state S_g of the engine is characterised by: (i) a state variable *temp* containing the current temperature of the engine; (ii) a sensor s_t (such as a thermometer or a thermocouple) measuring the temperature of the engine, (iii) an actuator *cool* to turn on/off the cooling system. The physical environment of the engine, Env_g, is constituted by: (i) a simple evolution law *evol* that increases (resp. decreases) the value of *temp*, when the cooling system is inactive (resp. active), by a value determined according to a discrete distribution of probability, taking into account an uncertainty in the model that may reach the threshold $\delta = 0.4$, and granularity g over reals; (ii) a measurement map *meas* returning the value detected by the sensor s_t determined by a discrete probability distribution based on a measurement error that may reach the threshold $err = 0.1$, and granularity g; (iii) an invariant set saying that the system gets faulty when the temperature of the engine gets out of the range $[0, 30]$.

Formally, $S_g = \langle \xi_\mathrm{x}, \xi_\mathrm{s}, \xi_\mathrm{a} \rangle$ and $Env_g = \langle evol, meas, inv \rangle$ with:

(i) $\xi_\mathrm{x} \in \mathbb{R}^{\{temp\}}$ and $\xi_\mathrm{x}(temp) = 0$;

(ii) $\xi_\mathrm{s} \in \mathbb{R}^{\{s_\mathrm{t}\}}$ and $\xi_\mathrm{s}(temp) = 0$;

(iii) $\xi_\mathrm{a} \in \mathbb{R}^{\{cool\}}$ and $\xi_\mathrm{a}(cool) = \mathsf{off}$; for the sake of simplicity, we can assume ξ_a to be a mapping $\{cool\} \to \{\mathsf{on}, \mathsf{off}\}$ such that $\xi_\mathrm{a}(cool) = \mathsf{off}$ if $\xi_\mathrm{a}(cool) \geq 0$, and $\xi_\mathrm{a}(cool) = \mathsf{on}$ if $\xi_\mathrm{a}(cool) < 0$.

Furthermore,

[3] Other forms of finite-support discrete probability distributions could be treated as well.

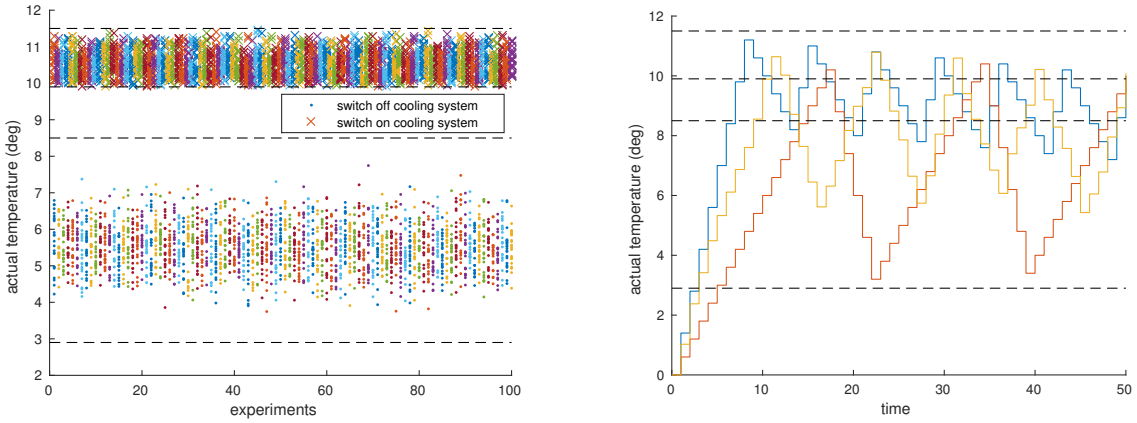

Figure 2: Simulations in MATLAB of the engine *Eng*

(i) $evol(\xi'_{\mathrm{x}}, \xi'_{\mathrm{a}}) = \sum_{v \in [v_1, v_2]_g} \frac{1}{|[v_1, v_2]_g|} \cdot \overline{[temp \mapsto \xi'_x(temp) + v]}$, for any $\xi'_x \in \mathbb{R}^{\{temp\}}$ and $\xi'_{\mathrm{a}} \in \mathbb{R}^{\{cool\}}$, where $[v_1, v_2] = [1{-}\delta\,,\ 1{+}\delta]$, if $\xi'_{\mathrm{a}}(cool) = \mathsf{off}$ (inactive cooling), and $[v_1, v_2] = [-1{-}\delta\,,\ -1{+}\delta]$, if $\xi'_{\mathrm{a}}(cool) = \mathsf{on}$ (active cooling);

(ii) $meas(\xi'_{\mathrm{x}}) = \sum_{v \in [-err, +err]_g} \frac{1}{|[-err, +err]_g|} \cdot \overline{[s_t \mapsto \xi'_x(temp) + v]}$, for any $\xi'_x \in \mathbb{R}^{\{temp\}}$;

(iii) $inv = \{[temp \mapsto x] \colon x \in \mathbb{R} \text{ and } 0 \leq x \leq 30\}$.

The *cyber component* of the engine consists of a process *Ctrl* which models the controller activity. Intuitively, process *Ctrl* senses the temperature of the engine at each time interval. When the sensed temperature is above 10, the controller activates the coolant. The cooling activity is maintained for 5 consecutive time units. After that time, if the temperature does not drop below 10 then the controller transmits its *ID* on a specific channel for signalling a *warning*, it keeps cooling for another 5 time units, and then checks again the sensed temperature; otherwise, if the sensed temperature is not above the threshold 10, the controller turns off the cooling and moves to the next time interval. Formally,

$$\begin{aligned} Ctrl &= \mathsf{rec}\, X.\mathsf{read}\, s_t(x).[x > 10]\{Cooling\}, \{\mathsf{tick}.X\} \\ Cooling &= \mathsf{write}\, cool\langle \mathsf{on}\rangle.\mathsf{rec}\, Y.\mathsf{tick}^5.\mathsf{read}\, s_t(x).[x > 10]\{\mathsf{snd}\, warning\langle \mathrm{ID}\rangle.Y\}, \{\mathsf{write}\, cool\langle \mathsf{off}\rangle.\mathsf{tick}.X\}\,. \end{aligned}$$

The *whole engine* is defined as: $Eng_g = Env_g; S_g \bowtie Ctrl$, where Env_g and S_g are the physical environment and the physical state defined before.

Our operational semantics allows us to formally prove a number of *run-time properties* of our engine. For instance, the following proposition says that our engine never reaches a warning state and never deadlocks.

Proposition 4.1. *Let Eng_g be the CPS defined before. Given any computation $Eng_g \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} M$, then $\alpha_i \in \{\tau, \mathsf{tick}\}$, for $1 \leq i \leq n$, and there is a distribution γ such that $M \xrightarrow{\alpha} \gamma$, for some $\alpha \in \{\tau, \mathsf{tick}\}$.*

Actually, knowing that in each of the 5 time slots of cooling, the temperature will drop of a value laying in the interval $[1{-}\delta, 1{+}\delta]_g$, we can be quite precise on the temperature reached by the engine before and after the cooling activity. Formally:

Proposition 4.2. *Let $Eng_g \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} M$ be an arbitrary computation of the engine, for some CPS M:*

- *if M turns the cooling on then the value of the state variable temp in M ranges over $(9.9, 11.5]$;*
- *if M turns the cooling off then the value of the variable temp in M ranges over $(2.9, 8.5]$.*

The proofs of both propositions can be found in the Appendix, in Section Appendix A.2.

The result formally proved in Proposition 4.2 finds a correspondence in the left graphic of Figure 2. In that graphic, we collect a campaign of 100 simulations of our engine in MATLAB[4], lasting 250 time units each, showing that the value of the state variable *temp* when the cooling system is turned on (resp., off) lays in the interval $(9.9, 11.5]$ (resp., $(2.9, 8.5]$); these bounds are represented by the dashed horizontal lines. Obviously, when dealing with complex systems even several thousands of simulations do not ensure the absence of incorrect states, as formally proved in Proposition 4.1 and Proposition 4.2.

The right graphic of the same figure shows three possible evolutions in time of the state variable *temp*: (i) the first one (in red), in which the temperature of the engine always grows of $1 - \delta = 0.6$ degrees per time step, when the cooling is off, and always decrease of $1 + \delta = 1.4$ degrees per time unit, when the cooling is on; (ii) the second one (in blue), in which the temperature always grows of $1 + \delta = 1.4$ degrees per time unit, when the cooling is off, and always decrease of $1 - \delta = 0.6$ degrees per time unit, when the cooling is on; (iii) and a third one (in yellow), in which, depending whether the cooling is off or on, at each time step the temperature grows or decreases of an arbitrary offset laying in the interval $[1 - \delta, 1 + \delta]$.

Now, the reader may wonder whether it is possible to design a variant of our engine which meets the same specification with better performances. For instance, an engine consuming less coolant. Let us consider a variant of the engine described before:

$$\widetilde{Eng_g} \;=\; \widetilde{Env_g}; S_g \bowtie Ctrl\,.$$

Here, $\widetilde{Env_g}$ is the same as Env_g except for the evolution map, as we set $[v_1, v_2] = [-0.8{-}\delta\,,\ -0.8{+}\delta]$ if $\xi'_{\mathrm{a}}(cool) = \mathsf{on}$ (active cooling). This means that in $\widetilde{Eng_g}$ we reduce the power of the cooling system by 20%. In Figure 3, we report the results of our simulations in MATLAB over 10000 runs lasting 10000 time units each. From this graph, $\widetilde{Eng_g}$ saves in average more than 10% of coolant with respect to Eng_g. So, the new question is: are these two engines behavioural equivalent? Do they meet the same specification?

Our bisimilarity provides us with a precise answer to these questions: the two variants of the engine are bisimilar.

Proposition 4.3. *$Eng_g \approx \widetilde{Eng_g}$, for any $g \in \mathbb{N}^+$.*

The proof can be found in the Appendix, in Section Appendix A.4.

At this point, one may wonder whether it is possible to improve the performances of our engine even more. For instance, by reducing the power of the cooling system by a further 10%, by setting $[v_1, v_2] = [-0.7{-}\delta\,,\ -0.7{+}\delta]$ if $\xi'_{\mathrm{a}}(cool) = \mathsf{on}$ (active cooling). We can formally prove that this is not possible.

Proposition 4.4. *Let $\widehat{Eng_g}$ be the same as Eng_g, except for the evolution map, in which the real interval $[v_1, v_2]$ is given by $[-0.7{-}\delta\,,\ -0.7{+}\delta]$ if $\xi'_{\mathrm{a}}(cool) = \mathsf{on}$. Then, $Eng_g \not\approx \widehat{Eng_g}$, for any $g \in \mathbb{N}^+$.*

The proof can be found in the Appendix, in Section Appendix A.2.

Finally, we show how we can use the compositionality of our behavioural semantics (Theorem 3.5) to deal with bigger CPSs. Suppose that Eng_g denotes the model in our calculus of an airplane engine. In this case, we could model a very simple *airplane control system* that checks whether the left engine (Eng_g^{L}) and the right engine (Eng_g^{R}) are signalling warnings. The whole CPS is defined as follows:

$$Airplane_g \;=\; ((Eng_g^{\mathrm{L}} \uplus Eng_g^{\mathrm{R}}) \parallel Check)\backslash\{warning\}$$

where $Eng_g^{\mathrm{L}} = Eng_g\{{}^{\mathrm{L}}/_{\mathrm{ID}}\}\{{}^{temp\_\mathrm{L}}/_{temp}\}\{{}^{cool\_\mathrm{L}}/_{cool}\}\{{}^{s_{\mathrm{t}}\_\mathrm{L}}/_{s_{\mathrm{t}}}\}$, and $Eng_g^{\mathrm{R}} = Eng_g\{{}^{\mathrm{R}}/_{\mathrm{ID}}\}\{{}^{temp\_\mathrm{R}}/_{temp}\}\{{}^{cool\_\mathrm{R}}/_{cool}\}\{{}^{s_{\mathrm{t}}\_\mathrm{R}}/_{s_{\mathrm{t}}}\}$, and process *Check* is defined as follows:

$$\begin{array}{rcl}
Check & = & \mathsf{rec}\,X.\lfloor \mathsf{rcv}\,warning(x).[x = \mathrm{L}]\{Check_1^{\mathrm{L}}\}, \{Check_1^{\mathrm{R}}\}\rfloor X \\
Check_i^{id} & = & \lfloor \mathsf{rcv}\,warning(y).[y \neq id]\{\mathsf{snd}\,alarm.\mathsf{tick}.X\}, \{\mathsf{tick}.Check_{i+1}^{id}\}\rfloor Check_{i+1}^{id} \\
Check_5^{id} & = & \lfloor \mathsf{rcv}\,warning(z).[z \neq id]\{\mathsf{snd}\,alarm.\mathsf{tick}.X\}, \{\mathsf{snd}\,failure\langle id\rangle.\mathsf{tick}.X\}\rfloor \\
 & & \mathsf{snd}\,failure\langle id\rangle.X
\end{array}$$

[4]MATLAB chooses a value in a real interval by means of a discrete uniform distribution depending on the granularity imposed by the finite number of bits used for the representation of floats.

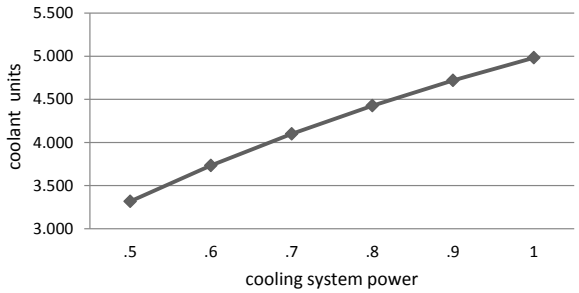

Figure 3: Simulations in MATLAB of coolant consumption

for $1 \leq i \leq 5$. Intuitively, if one of the two engines is in a warning state then the process $Check_i^{id}$, for $id \in \{\mathrm{L}, \mathrm{R}\}$, checks whether also the second engine moves into a warning state, in the following 5 time intervals (i.e. during the cooling cycle). If both engines get in a warning state then an *alarm* is sent, otherwise, if only one engine is facing a warning then the airplane control system yields a *failure* signalling which engine is not working properly.

So, since we know that $Eng_g \approx \widetilde{Eng_g}$, for any $g \in \mathbb{N}^+$, the final question becomes the following: can we safely equip our airplane with the more performant engines, $\widetilde{Eng_g^{\mathrm{L}}}$ and $\widetilde{Eng_g^{\mathrm{R}}}$, in which $[v_1, v_2] = [-0.8{-}\delta\,,\ -0.8{+}\delta]$, if $\xi'_{\mathrm{a}}(cool) = \mathsf{on}$, without affecting the whole observable behaviour of the airplane? The answer is "yes", and this result can be formally proved by relying on Proposition 4.3 and Theorem 3.5.

Proposition 4.5. *Let* $\widetilde{Airplane_g} = ((\widetilde{Eng_g^{\mathrm{L}}} \uplus \widetilde{Eng_g^{\mathrm{R}}}) \parallel Check)\backslash\{warning\}$. *Then,* $Airplane_g \approx \widetilde{Airplane_g}$.

We end this section with an observation. Although, the engine $\widehat{Eng_g}$ is not behavioural equivalent to the original engine Eng_g, an airplane maker might be interested in knowing an estimation of the deviation of its behaviour with respect to the behaviour of the original engine. If this deviation would be very small then aeronautical engineers might consider to adopt in their airplanes the engine $\widehat{Eng_g}$ instead $\widetilde{Eng_g}$ to save even more coolant. So, the new question is: how big is the deviation, in terms of behaviour, of the engine $\widehat{Eng_g}$ with respect to the original engine Eng_g?

The rest of the paper is devoted to develop general quantitative techniques to estimate the deviation of the probabilistic behaviour of a CPS with respect to another.

5. Bisimulation metrics

In this section, we provide a weak behavioural distance to compare the probabilistic behaviour of CPSs up to a given approximation. To this end, we adapt the notion of *weak bisimilarity metric* [23] to pCCPS. Intuitively, we will write $M \approx_p N$ if the weak bisimilarity between M and N holds with a *distance* p, with $p \in [0, 1]$. Thus, $\approx_0$ will coincide with the weak probabilistic bisimilarity of Definition 3.2, whereas $\bigcup_{p\in[0,1]} \approx_p$ will correspond to the cartesian product $\mathtt{pCCPS} \times \mathtt{pCCPS}$.

Weak bisimilarity metric is defined as a *pseudometric* measuring the tolerance of the probabilistic weak bisimilarity.

Definition 5.1 (Pseudometric)**.** *A function* $d\colon \mathtt{pCCPS} \times \mathtt{pCCPS} \to [0, 1]$ *is said to be a* 1-bounded pseudometric *if*

- $d(M, M) = 0$, *for all* $M \in \mathtt{pCCPS}$,
- $d(M, M') = d(M', M)$, *for all* $M, M' \in \mathtt{pCCPS}$,
- $d(M, M') \leq d(M, M'') + d(M'', M')$, *for all* $M, M', M'' \in \mathtt{pCCPS}$.

Weak bisimilarity metric provides the quantitative analogous of the weak bisimulation game: two CPSs M and N at distance p can mimic each other's transitions and evolve to distributions γ and γ', respectively, placed at some distance q, with $q \leq p$. This requires to lift pseudometrics from CPSs to distributions over CPSs. To this end, as in [34, 35], we rely on the notions of *matching* [37] and *Kantorovich lifting* [39][5].

[5]The original formulation of weak bisimulation metric [23] is technically different but equivalent to our definition [40].

In Definition 3.1, we already provided the definition of matching. Let us define the Kantorovich lifting.

Definition 5.2 (Kantorovich lifting)**.** *Let $d\colon \mathtt{pCCPS} \times \mathtt{pCCPS} \to [0,1]$ be a pseudometric. The* Kantorovich lifting *of d is the function $\mathbf{K}(d)\colon \mathcal{D}(\mathtt{pCCPS}) \times \mathcal{D}(\mathtt{pCCPS}) \to [0,1]$ defined as:*

$$\mathbf{K}(d)(\gamma,\gamma') \;=\; \min_{\omega\in\Omega(\gamma,\gamma')} \sum_{M,M'\in \mathtt{pCCPS}} \omega(M,M')\cdot d(M,M')$$

for all $\gamma,\gamma' \in \mathcal{D}(\mathtt{pCCPS})$.

Note that since we are considering only distributions with finite support, the minimum over the set of matchings $\Omega(\gamma,\gamma')$ is well defined.

Definition 5.3 (Weak bisimulation metric)**.** *We say that a pseudometric $d\colon \mathtt{pCCPS} \times \mathtt{pCCPS} \to [0,1]$ is a* weak bisimulation metric *if for all $M,N \in \mathtt{pCCPS}$, with $d(M,N) < 1$, whenever $M \xrightarrow{\alpha} \gamma$ there is a sub-distribution γ' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} \gamma'$ and $\mathbf{K}(d)(\gamma\,,\,\gamma' + (1-|\gamma'|)\overline{\mathsf{Dead}}) \le d(M,N)$.*

Note that in the previous definition, if $|\gamma'|< 1$ then, with probability $1-|\gamma'|$, there is no way to simulate the behaviour of any CPS with a valid invariant in the support of γ (the special CPS Dead does not perform any action).

A crucial result is the existence of the minimal weak bisimulation metric [23], called *weak bisimilarity metric*, and denoted with $\mathbf{d}$. We remark that in [23] it is shown that the kernel of $\mathbf{d}$ coincides with the definition of weak probabilistic bisimilarity.

Proposition 5.4. *For all $M,N \in \mathtt{pCCPS}$ we have $\mathbf{d}(M,N) = 0$ if and only if $M \approx N$.*

Now, we have all ingredients to define our notion of behavioural distance between CPSs.

Definition 5.5 (Distance between CPSs)**.** *Let $M,N \in \mathtt{pCCPS}$ and $p \in [0,1]$. We say that M* and *N* have distance *p, written $M \approx_p N$, if and only if $\mathbf{d}(M,N) = p$.*

In the next section, we will use a more refined notion of distance that considers only the first $n \in \mathbb{N}$ computation steps, when comparing two CPSs.

Such definition requires the introduction of a complete lattice $([0,1]^{\mathtt{pCCPS}\times\mathtt{pCCPS}}, \sqsubseteq)$ of functions of type $\mathtt{pCCPS} \times \mathtt{pCCPS} \to [0,1]$ ordered by $d_1 \sqsubseteq d_2$ iff $d_1(M,N) \le d_2(M,N)$ for all $M,N \in \mathtt{pCCPS}$, where for each set $D \subseteq [0,1]^{\mathtt{pCCPS}\times\mathtt{pCCPS}}$ the supremum and infimum are defined as $\sup(D)(M,N) = \sup_{d\in D} d(M,N)$ and $\inf(D)(M,N) = \inf_{d\in D} d(M,N)$, for all $M,N \in \mathtt{pCCPS}$. Notice that the infimum of the lattice is the constant function zero, which we denote by $\mathbf{0}$.

We also need a functional $\mathbf{B}$ defined over the lattice mentioned above such that $\mathbf{B}(d)(M,N)$ returns the minimum possible value for $d(M,N)$ in order to ensure that d is a weak bisimulation metric.

Definition 5.6 (Bisimulation metric functional)**.** *Let $\mathbf{B}\colon [0,1]^{\mathtt{pCCPS}\times\mathtt{pCCPS}} \to [0,1]^{\mathtt{pCCPS}\times\mathtt{pCCPS}}$ be the functional such that for any $d \in [0,1]^{\mathtt{pCCPS}\times\mathtt{pCCPS}}$ and $M,N \in \mathtt{pCCPS}$, $\mathbf{B}(d)(M,N)$ is given by:*

$$\sup_{\{\alpha\,:\,M\xrightarrow{\alpha}\,\vee\, N\xrightarrow{\alpha}\}} \max\left\{ \max_{M\xrightarrow{\alpha}\gamma_1} \min_{N\stackrel{\hat{\alpha}}{\Longrightarrow}\gamma_2} \mathbf{K}(d)(\gamma_1, \gamma_2 + (1-|\gamma_2|)\overline{\mathsf{Dead}}),\; \max_{N\xrightarrow{\alpha}\gamma_2} \min_{M\stackrel{\hat{\alpha}}{\Longrightarrow}\gamma_1} \mathbf{K}(d)(\gamma_1 + (1-|\gamma_1|)\overline{\mathsf{Dead}}, \gamma_2) \right\}$$

where $\max\emptyset = 0$ and $\min\emptyset = 1$.

Notice that Definition 5.6 and Definition 5.3 are strictly related as weak bisimulation metrics are pseudometrics that are prefixed points of $\mathbf{B}$. Notice also that all max and min in Definition 5.6 are well defined since our pLTS is image-finite and CPSs enjoy the well timedness property.

Since $\mathbf{K}$ is monotone [41] it follows that $\mathbf{B}$ is a monotone function on $([0,1]^{\mathtt{pCCPS}\times\mathtt{pCCPS}}, \sqsubseteq)$. Furthermore, since this structure is a lattice, by Knaster-Tarski theorem it follows that $\mathbf{B}$ has a least prefixed point (which is also the least fixed point). Later we will show that this least prefixed point coincides with $\mathbf{d}$.

Now, we exploit the functional $\mathbf{B}$ to introduce a notion of *n-weak bisimilarity metric*, denoted $\mathbf{d}^n$, which intuitively quantifies the tolerance of the weak bisimulation in n steps. The idea is that $\mathbf{d}^0$ coincides with the constant function $\mathbf{0}$ assigning distance 0 to all pairs of CPSs, whereas $\mathbf{d}^n(M,N)$, for $n > 0$, is defined as $\mathbf{d}^n(M,N) = \mathbf{B}(\mathbf{d}^{n-1})(M,N)$. Thus, the n-weak bisimilarity metric between M and N is defined in terms of the $(n-1)$-weak bisimilarity metric between the distributions reached (in one step) by M and N, respectively.

Definition 5.7 (n-weak bisimilarity metric)**.** *Let $n \in \mathbb{N}$. The function $\mathbf{B}^n(\mathbf{0})$, abbreviated as $\mathbf{d}^n$, is called* n-weak bisimilarity metric.

Proposition 5.8. *For all $n \geq 0$, $\mathbf{d}^n$ is a 1-bounded pseudometric.*

The proof of this proposition can be found in Appendix, in Section Appendix A.3.

Finally, we are ready to define our notion of n-distance between two CPSs.

Definition 5.9 (n-distance between CPSs)**.** *Let $M, N \in \mathtt{pCCPS}$ and $p \in [0, 1]$. We say that M* and *N* have *n*-distance *p, written $M \approx_p^n N$, if and only if $\mathbf{d}^n(M, N) = p$.*

Since our pLTS is image-finite, and all transitions lead to distributions with finite support, it is possible to prove that $\mathbf{B}$ is continuous [42]. Since $\mathbf{B}$ is also monotone, we can deduce that the closure ordinal of $\mathbf{B}$ is ω (see Section 3 of [42]). As a consequence, the n-weak bisimilarity metrics converge to the weak bisimilarity metric when n grows indefinitely. Formally,

Proposition 5.10. $\mathbf{d} = \lim_{n\to\infty} \mathbf{d}^n$.

Last but but not least, the distances introduced in Definition 5.5 and Definition 5.9 allow us to compare CPSs in a compositional manner. In particular, these distances are preserved by parallel composition of physical-disjoint CPSs, by parallel composition of pure-logical processes, and by channel restriction.

Theorem 5.11 (Compositionality of distances)**.** *Let M and N be two arbitrary CPSs in $\mathtt{pCCPS}$.*

1. *$M \approx_p N$ implies $M \uplus O \approx_q N \uplus O$, with $q \leq p$, for any physically-disjoint CPS O;*
2. *$M \approx_p N$ implies $M \parallel P \approx_q N \parallel P$, with $q \leq p$, for any pure-logical process P;*
3. *$M \approx_p N$ implies $M\backslash c \approx_q M\backslash c$, with $q \leq p$, for any channel c;*
4. *$M \approx_p^n N$ implies $M \uplus O \approx_q^n N \uplus O$, with $q \leq p$, for any physically-disjoint CPS O and any $n \geq 0$;*
5. *$M \approx_p^n N$ implies $M \parallel P \approx_q^n N \parallel P$, with $q \leq p$, for any pure-logical process P and any $n \geq 0$;*
6. *$M \approx_p^n N$ implies $M\backslash c \approx_q^n M\backslash c$, with $q \leq p$, for any channel c and $n \geq 0$.*

The proof of Theorem 5.11 can be found in the Appendix, in Section Appendix A.3.

Now, suppose that $M \approx_p N$, $M' \approx_{p'} N'$, with M (resp. N) and M' (resp. N') physically-disjoint. By Theorem 5.11.1, we can infer both $M \uplus M' \approx_q N \uplus M'$ and $N \uplus M' \approx_{q'} N \uplus N'$, with $q \leq p$ and $q' \leq p'$. Then, by triangular property of the pseudometric $\mathbf{d}$ we get $M \uplus M' \approx_{q''} N \uplus N'$, for some $q'' \leq q + q' \leq p + p'$. Similarly, by applying Theorem 5.11.4 we can infer that $M \approx_p^n N$ and $M' \approx_{p'}^n N'$ entail $M \uplus M' \approx_q^n N \uplus N'$, for some $q \leq p + p'$. This says that our metrics enjoy a well-known compositional property called *non-expansiveness* [24, 43, 44].

In the next section, the compositional properties of Theorem 5.11 will be very useful when reasoning on our case study.

6. Case study, reloaded

In Section 4, we proved that the original version of the proposed engine, Eng_g, and its variant $\widetilde{Eng_g}$ (saving up to 10% of coolant) are behavioural equivalent (i.e., bisimilar). Then, by relying on the compositionality of our probabilistic bisimilarity (Theorem 3.5), we proved that the two compound systems, $Airplane_g$ and $\widetilde{Airplane}_g$, mounting engines Eng_g and $\widetilde{Eng_g}$, respectively, are bisimilar as well.

Actually, both results can be proved in terms of weak probabilistic metric with distance 0, as this specific metric coincides with the probabilistic bisimilarity (Proposition 5.4).

Proposition 6.1. *Let $g \in \mathbb{N}^+$. Then,*

- $Eng_g \approx_0 \widetilde{Eng_g}$

- $Airplane_g \approx_0 \widetilde{Airplane}_g$.

Then, in Section 4 we moved our attention to a more performant engine, $\widehat{Eng_g}$, saving almost 20% of coolant with respect to the original engine Eng_g. In our behavioural analysis we rejected this new variant as it may exhibit a different probabilistic behaviour when compared to Eng_g. More precisely, the two systems Eng_g and $\widehat{Eng_g}$ *are not* bisimilar (Proposition 4.4).

However, in many complex probabilistic systems, such as CPSs, probabilistic bisimilarity might reveal to be too strong as the natural behavioural equivalence to take systems apart. Thus, in Section 4 we advocated for some appropriate notion of behavioural distance to estimate the effective difference, in terms of behaviour, of these two versions of the engine.

In the current section, we apply the bisimulation metrics defined in Section 5 to estimate the distance between Eng_g and $\widehat{Eng_g}$, by varying the granularity $g \in \mathbb{N}^+$. In particular, we apply the notion of n-weak bisimilarity metric.

Proposition 6.2. *Let $g \in \mathbb{N}^+$ and $n \in \mathbb{N}$. Then, for $p_g = \frac{|[0.3,0.4)_g|}{|[0.3,1.1]_g|}$ and $q_g = \frac{|(1.3,1.4]_g|}{|[0.6,1.4]_g|}$, we have:*

$$\mathbf{d}^n(Eng_g, \widehat{Eng_g}) \;\leq\; 1 - \left(1 - q_g(p_g)^5\right)^n \quad .$$

Note that if the cooling system of $\widehat{Eng_g}$ is off and it is not going to be activated in the current time slot, then the sensed temperature is below than or equal to 10, and the real temperature is below than or equal to 10.1 degrees (we recall that $err = 0.1$). Assume that the temperature is exactly 10.1. If in the current time slot the temperature increases of a value $v \in (1.3, 1.4]$ then it will reach a value in the interval $(11.4, 11.5]$ (we recall that $\delta = 0.4$). This happens with a probability bounded by q_g. In this case, the cooling system will be turned on, and the temperature will drop, in each of the following 5 time slots, of some value laying in the interval $[0.7{-}\delta\,,\ 0.7{+}\delta] = [0.3, 1.1]$. However, if in each of those 5 slots of cooling the temperature is decreased of a value laying in $[0.3, 0, 4)$, then the cooling activity might not be enough to avoid (observable) warnings, and the two engines Eng_g and $\widehat{Eng_g}$ will be distinguished. Thus, p_g is given by the number of possible "bad decreases", $|\ [0.3, 0.4)_g\ |$, divided by the number of all possible decreases, $|\ [0.3, 1.1]_g\ |$; whereas q_g is given by the number of possible "bad increases", $|\ (1.3, 1.4]_g\ |$, divided by the number of all possible increases $|\ [0.6, 1.4]_g\ |$.

Notice that p_g and q_g refer to real intervals which are basically shifted. Thus, we have that $|\ [0.3, 0.4)_g\ | = |\ (1.3, 1.4]_g\ | = 10^{g-1}$ and $|\ [0.3, 1.1]_g\ | = |\ [0.6, 1.4]_g\ | = 8 \cdot 10^{g-1} + 1$. As a consequence, $p_g = q_g = \frac{10^{g-1}}{8 \cdot 10^{g-1}+1} = \frac{1}{8+10^{-g+1}}$. Obviously, the finer is the granularity g the closer is the value of p_g and q_g to $\frac{1}{8}$. Formally,

$$\lim_{g\to\infty} \mathbf{d}^n(Eng_g, \widehat{Eng_g}) \;\leq\; 1 - (1 - \frac{1}{8^6})^n \,. \tag{1}$$

Thus, for instance, assuming a granularity $g = 6$, after $n = 3000$ computation steps the distance between the two systems is less than 0.012. Intuitively, this means that if we limit our analysis to 3000 computation steps the behaviours of two engines may differ with probability at most 0.012. By an easy inspection in the (common) logics of the two engines, it is easy to see that any two subsequent tick-actions are separated by at most 2 untimed actions. Thus, 3000 computation steps means around 1000 time slots. Considering time slots lasting 20 seconds each, this means more than five hours. Thus, an utilisation of $\widehat{Eng_g}$ might be feasible in airplanes used for short-range flights, where the engine is actually used for a limited amount of time. Actually, aeronautical engineers might consider perfectly acceptable the risk of mounting the engine $\widehat{Eng_g}$ instead of Eng_g, when compared to the reliability of the other components of the airplane.

However, since an airplane mounts two engines, engineers need to estimate the difference in terms of behaviour on the whole airplane resulting by the adoption of different versions of the engine. This is exactly the point where we can rely on Theorem 5.11 to support compositional reasoning.

The following result follows from Equation 1, Proposition 6.2 and Theorem 5.11.

Proposition 6.3. *Let $g \in \mathbb{N}^+$ and $n \in \mathbb{N}$. Let $\widehat{Airplane}_g \;=\; ((\widehat{Eng_g^{\mathrm{L}}} \uplus (\widehat{Eng_g^{\mathrm{R}}}) \parallel Check)\backslash\{warning\}$. Then,*

1. $\mathbf{d}^n(Airplane_g\ ,\ \widehat{Airplane}_g) \;\leq\; 2p$, *where* $p = 1 - (1 - q_g(p_g)^5)^n$

2. $\lim_{g\to\infty} \mathbf{d}^n(Airplane_g\ ,\ \widehat{Airplane}_g) \ \leq\ 2(1-\left(1-\frac{1}{8^6}\right)^n)$.

Thus, for $g = 6$, the probability that the two airplanes mounting different engines exhibit a different behaviour within $n = 3000$ computation steps is at most 0.024; a distance which may be considered still acceptable in specific contexts. Notice that in the (common) logics of the two airplanes, it is easy to see that two tick-actions are separated by at most 5 untimed actions (two for each engine plus one to signal a possible alarm). Thus, 3000 computation steps means around 600 time slots, i.e., more than three hours for time slots lasting 20 second each.

Finally, the reader should notice that the bound of the distance between the two airplanes is given by the summation of the bounds of the distances between the two corresponding engines. This is perfectly in line with the fact that our bisimulation metrics enjoy the *non-expansiveness* property.

The proofs of the previous propositions can be found in the Appendix, in Section Appendix A.4.

7. Conclusions, related and future work

We have proposed a hybrid probabilistic process calculus, called pCCPS, for specifying and reasoning on cyber-physical systems. Our calculus allows us to model a CPS by specifying its *physical plant*, containing information on state variables, sensors, actuators, evolution law, etc., and its *logics*, i.e., controllers, IDSs, supervisors, etc. Physical and logical components interact through sensors and actuators, whereas interactions within the logics or between logics of different CPSs rely on channel-based communication. In pCCPS, the representation of the evolution map takes into account the uncertainty of the physical model, whereas the representation of the measurement map considers measurement errors in sensor reading. As a consequence, the two maps returns discrete probability distributions over state functions and sensor functions, respectively.

pCCPS is equipped with a probabilistic labelled transition semantics which satisfies classical time properties: *time determinism*, *patience*, *maximal progress*, and *well-timedness*. As behavioural semantics we adopt a natural notion of *weak probabilistic bisimilarity* which is proved to be preserved by appropriate system contexts that are suitable for *compositional reasoning*. Then, we argue that probabilistic bisimilarity is only partially satisfactory to reason on CPSs as it can only establish whether two CPSs behave exactly in the same way. To this end, we generalise our probabilistic bisimilarity to provide a notion of *weak bisimulation metric* along the lines of [23]. We also define a notion of weak bisimulation metric in n steps, which reveals to be very effective whenever it is not necessary to observe the system "ad infinitum" but it is enough to observe its behaviour restricted to bounded computations. Again, both bisimulation metrics are proved to be suitable for compositional reasonings. The paper provides a case study, taken from an engineering application, and uses it to illustrate our definitions and our compositional probabilistic behavioural theory for pCCPS.

Related work. A number of *hybrid process algebras* [9, 10, 11, 12, 13] have been proposed for reasoning about physical systems and provide techniques for analysing and verifying protocols for hybrid automata. Among these approaches, pCCPS shares some similarities with the ϕ-calculus [12], a hybrid extension of the π-calculus [6] equipped with a weak bisimilarity that is not compositional. Galpin et al. [13] proposed a process algebra, called HYPE, in which the continuous part of the system is represented by appropriate variables whose changes are determined by active influences (i.e., commands on actuators). The authors define a strong bisimulation that extends the *ic-bisimulation* of [10]. Unlike ic-bisimulation, the bisimulation in HYPE is preserved by a notion of parallel composition that is slightly more permissive than ours. However, bisimilar systems in HYPE must always have the same influence. Thus, in HYPE we cannot compare CPSs sending different commands on actuators at the same time, as we do (for instance) in Proposition 4.3.

Several approaches have been proposed in the last years [14, 15, 16, 17, 18, 19, 20] to enrich hybrid models with probabilistic or stochastic behaviour. Most of them consist in introducing either probabilities in the transition relation, or probabilistic choice, or stochastic differential equations. For instance, in *Stochastic Hybrid CSP* (SHCSP) [20] probabilistic choice replaces non-deterministic choice, stochastic differential equations replace differential equations, and communication interrupts are generalised by communication interrupts with weights.

The formal analysis of probabilistic and stochastic systems follows the two classic mainstreams: (i) *model checking* (e.g., [17]) and *reachability* (e.g., [14, 17]), when the focus is on a single system; (ii) *behavioural equivalences* (e.g., [45, 22, 38, 46, 47, 48]) when the goal is to compare the behaviour of two systems (very often, specification and

implementation of the same system). As already said in the Introduction, probabilistic behavioural equivalences may be too strong in certain probabilistic and stochastic models in which many interesting systems are only approximately behavioural equivalent. This led to several notions of *behavioural distance* that can be grouped in two main families: quantitative counterparts of trace equivalence [49, 50, 51, 52], and quantitative counterparts of bisimulation equivalence [23, 24, 25, 26]. We refer to [53, 54] for a comparison between these two approaches. In the present paper, we have adopted a bisimulation-based definition because, unlike trace semantics, bisimulation is sensitive to system deadlock, a phenomenon that has a great impact in CPSs.

More generally, we are aware of a number of works using formal methods for studying CPSs or IoT systems, although they apply methods, and most of the time have goals, that are quite different from ours.

Vigo et al. [55] proposed a calculus for wireless-based cyber-physical systems endowed with a theory to study cryptographic primitives, together with explicit notions of communication failure and unwanted communication. The calculus does not provide any notion of behavioural equivalence. It also lacks a clear distinction between physical and logical components. Lanese et al. [56] proposed an untimed calculus of mobile IoT devices interacting with the physical environment by means of sensors and actuators. The calculus does not allow any representation of the physical environment, and it is equipped with an end-user bisimilarity in which end-users may: (i) provide values to sensors, (ii) check actuators, and (iii) observe the mobility of smart devices. End-user bisimilarity is not preserved by parallel composition. Compositionality is recovered by strengthening its discriminating power. Lanotte and Merro [57] extended and generalised the work of [56] in a timed setting by providing a bisimulation-based semantic theory that is suitable for compositional reasoning. As in [56], the physical environment is not represented. Bodei et al. [58] proposed a new untimed process calculus, IoT-LYSA, supporting a control flow analysis that safely approximates the abstract behaviour of IoT systems. Essentially, they track how data spread from sensors to the logics of the network, and how physical data are manipulated. The calculus adopts asynchronous multi-party communication among nodes taking care of node proximity (the topology is static). The dynamics of the calculus is given in terms of a reduction relation. No behavioural equivalences are defined.

Finally, the paper at hand extends the conference paper [1] in the following aspects: (i) the calculus has become a probabilistic calculus, both in its logical and its physical components; the logics has been enriched with probabilistic choice, whereas discrete (finite-support) probability distributions have replaced continuous non-deterministic uncertainties in the evolution and continuous non-deterministic error-prone measurements; (ii) standard bisimulation has been replaced with probabilistic bisimulation and then with bisimulation metrics; (iii) as a consequence, the case study has been revisited using our bisimulation metrics to estimate the deviation in terms of behaviour of the systems under investigation.

Current and future work. We believe that our paper can lay and streamline *theoretical foundations* for the development of formal and automated tools to verify CPSs before their practical implementation. To that end, we will consider applying, possibly after proper enhancements, existing tools and frameworks for automated verification, such as Maude [59], PRISM [5], SMC UPPAAL [60] and Ariadne [61], resorting to the development of a dedicated tool if existing ones prove not up to the task. We are currently working [62] on a non-probabilistic version of pCCPS extended with security features to provide a formal study of a variety of *cyber-physical attacks* targeting physical devices of CPSs. In a second paper [63], we have recently extended and generalised our notion of n-weak bisimulation metric to focus on timed actions only. This allowed us to provide a compositional metric to estimate the impact of cyber-physical attacks on sensor devices with a special care on the time aspects of attacks.

As possible future work, a non-trivial challenge would be to extend the present work in order to deal with *continuous probability distributions*. In our setting, this would mean, for instance, that the evolution map *evol* should return a continuous distribution over state functions, and that the function $next_E(S)$ should return a continuous distributions over physical states. However, this would immediately give rise to a serious technical problem: the definition of probabilistic *weak labelled transitions*, and hence the definition of *weak behavioural equivalences and distances*. To better illustrate the problem, suppose to adopt continuous probability distributions in our calculus, and suppose a cyber-physical system M such that $M \xrightarrow{\mathsf{tick}} \gamma$, for some continuous probability distribution γ over CPSs. Suppose γ is a *uniform distribution* such that $\mathsf{supp}(\gamma) = \{M_r : r \in [0,1]\}$, with $M_r \neq M_{r'}$, for any $r \neq r'$. Independently on the specific definition of the CPSs M_r, as the logics of any CPS is intrinsically discrete, the cyber-component of any M_r will drive the whole system to a *discrete distribution*. As an example, assume a cyber-physical system N such that for all reals $r \in [0, 0.5]$ there

is a τ-transition $M_r \xrightarrow{\tau} \overline{N}$; whereas for all reals $r \in (0.5, 1]$ there is a τ-transition $M_r \xrightarrow{\tau} \overline{M_r}$. In such a situation, it is far from obvious to determine what should be the distribution γ_m reached by the original CPS M after a weak tick-transition, $M \stackrel{\mathsf{tick}}{\Longrightarrow} \gamma_m$. In fact, γ_m can be neither a discrete nor a continuous distribution. This because γ_m should map N to a probability weight 0.5 (as in a discrete distribution), and then it should distribute the remaining mass probability as a uniform (sub-)distribution to all M_r with $r \in (0.5, 1]$, such that $\int_{0.5}^{1} \gamma_m(M_t)dt = 0.5$ (as in a continuous distribution). Some preliminary work in this direction has been recently proposed in [64].

A possible solution to capture weak transitions when working with continuous probability distributions is to approximate them via discrete ones by adopting the approach proposed for labelled Markov processes in [65, 24]. In these papers, Desharnais et al. propose approximation techniques for continuous-state labelled Markov processes $\mathcal{S}$ in terms of finite-state Markov chains $\mathcal{S}(n, \epsilon)$, parametric in a natural number n and a rational number $\epsilon > 0$. Here, n is the maximal number of possible consecutive transitions from the start state of $\mathcal{S}(n, \epsilon)$ (the idea being that this Markov chain is the n-steps unfolding of the original Markov process $\mathcal{S}$), whereas the rational number $\epsilon > 0$ measures the accuracy of probabilities in $\mathcal{S}(n, \epsilon)$ when approximating the transitions of the original process $\mathcal{S}$. In their Theorem 4.4 [65] the authors prove that if a state s of $\mathcal{S}$ satisfies a formula in the logic characterising probabilistic bisimulation then there is some approximation $\mathcal{S}(n, \epsilon)$ satisfying exactly the same formula. Furthermore, the same authors show that one can always reconstruct the original process from the approximations. More precisely, a Markov process bisimilar to the original one can always be derived from the countable approximates $\mathcal{S}(n, 2^{-n})$, for some $n \in \mathbb{N}$ (in the current paper we adopted a granularity $\epsilon = 10^{-n}$). Actually, they do not reconstruct the original state space, but they reconstruct all the transition probability information, i.e., the dynamical aspects of the process (see Theorem 4.5 of [65]).

Acknowledgements. We thank the anonymous reviewers for their insightful and careful reviews.

Appendix A. Proofs

Appendix A.1. Proofs of Section 2

Theorem 2.9 states that CPSs enjoy time determinism, maximal progress, patience and well-timedness. We start by showing that processes enjoy the same properties.

Lemma Appendix A.1 (Processes time properties)**.** *Let P be a process of $\mathtt{pCCPS}$.*

(a) If $P \xrightarrow{\mathsf{tick}} \pi$ and $P \xrightarrow{\mathsf{tick}} \pi'$, then $\pi \equiv \pi'$.

(b) If $P \xrightarrow{\tau} \pi$ then there is no π' such that $P \xrightarrow{\mathsf{tick}} \pi'$.

(c) If $P \xrightarrow{\mathsf{tick}} \pi'$ for no π' then there is π such that $P \xrightarrow{\lambda} \pi$ for some $\lambda \in \{\tau, a!v, s?(x)\}$.

(d) There is a $k \in \mathbb{N}$ such that if $P \xrightarrow{\lambda_1} \cdots \xrightarrow{\lambda_n} P'$, with $\lambda_i \neq \mathsf{tick}$, then $n \leq k$.

Proof. We show the four properties separately.

(a) The proof is by induction on the depth d of the derivation tree allowing us to derive $P \xrightarrow{\mathsf{tick}} \pi$.

Base case $d = 1$. The transition $P \xrightarrow{\mathsf{tick}} \pi$ is derived by applying one of the rules (TimeNil), (Delay) and (Timeout), and the thesis is immediate.

Inductive case $d > 1$. The transition $P \xrightarrow{\mathsf{tick}} \pi$ is derived by applying one of the rules (TimePar), (ChnRes) and (Rec). We consider the case (TimePar), the others are similar. Since $P \xrightarrow{\mathsf{tick}} \pi$ is derived by rule (TimePar), process P must be of the form $P \equiv P_1 \parallel P_2$ for suitable processes P_1 and P_2. Therefore also the rule $P \xrightarrow{\mathsf{tick}} \pi'$ is derived through rule (TimePar). We have

$$\frac{P_1 \xrightarrow{\mathsf{tick}} \pi_1 \quad P_2 \xrightarrow{\mathsf{tick}} \pi_2 \quad P_1 \parallel P_2 \xrightarrow{\tau}\!\!\!\!\!/}{P_1 \parallel P_2 \xrightarrow{\mathsf{tick}} \pi_1 \parallel \pi_2} \qquad \frac{P_1 \xrightarrow{\mathsf{tick}} \pi'_1 \quad P_2 \xrightarrow{\mathsf{tick}} \pi'_2 \quad P_1 \parallel P_2 \xrightarrow{\tau}\!\!\!\!\!/}{P_1 \parallel P_2 \xrightarrow{\mathsf{tick}} \pi'_1 \parallel \pi'_2}$$

with $\pi = \pi_1 \parallel \pi_2$ and $\pi' = \pi'_1 \parallel \pi'_2$.

By the inductive hypothesis we have that $\pi_1 \equiv \pi'_1$ and $\pi_2 \equiv \pi'_2$, which gives $\pi_1 \parallel \pi_2 \equiv \pi'_1 \parallel \pi'_2$ and concludes the proof.

(b) The proof is by induction on the depth d of the derivation tree allowing us to derive $P \xrightarrow{\tau} \pi$.

Base case $d = 1$. There is no rule in Table 1 allowing us to derive transition $P \xrightarrow{\tau} \pi$ with depth 1, hence the thesis follows trivially.

Inductive case $d > 1$. The transition $P \xrightarrow{\tau} \pi$ is derived by applying one of the rules (Com), (Par), (ChnRes) and (Rec). We consider the case (Com). Since $P \xrightarrow{\tau} \pi$ is derived by rule (Com), process P must be of the form $P \equiv P_1 \parallel P_2$ for suitable processes P_1 and P_2. To show the thesis that no transition from $P_1 \parallel P_2$ labelled tick can be derived, it is enough to note that the only rule in Table 1 which may be applied to infer any tick-labelled transition from $P_1 \parallel P_2$ is rule (TimePar), which cannot be applied since it has $P_1 \parallel P_2 \xrightarrow{\tau}\!\!\!\!\!/\;$ among its premises.

The other cases follow directly by induction.

(c) First of all we notice that, if $P = \mathsf{rec}\, X.Q$, then, since P is bounded and has time-guarded recursion, by applying repetitively the structural congruence $\mathsf{rec}\, X.Q \equiv Q\{^{\mathsf{rec}\, X.Q}/_X\}$, we find a process $P' \equiv P$ such that $P' \neq \mathsf{rec}\, Y.R$, for any Y and R. Since $P' \equiv P$ implies $P' \xrightarrow{\lambda}$ iff $P \xrightarrow{\lambda}$, for any λ, we can prove the thesis by structural induction on P where P is not of the form $P = \mathsf{rec}\, X.Q$.

The base cases $P = \mathsf{nil}$, $P = \mathsf{tick}.C$ and $P = \lfloor chn.C \rfloor D$ are immediate since in all these cases a transition labelled tick from P can be derived. The base case $P = phy.C$ holds since we can apply either rule (Write) to derive a transition from P labelled $a!v$, or rule (Read) to derive a transition labelled $s?(x)$.

The inductive steps are $P = P_1 \parallel P_2$, $P = [b]\{P_1\},\{P_2\}$ and $P = Q\backslash c$. Consider the case $P = P_1 \parallel P_2$. If no transition from $P_1 \parallel P_2$ labelled tick can be derived, then rule (TimePar) cannot be applied. Then, at least one of the premises $P_1 \xrightarrow{\mathsf{tick}} \pi_1$, $P_2 \xrightarrow{\mathsf{tick}} \pi_2$ and $P_1 \parallel P_2 \xrightarrow{\tau}\!\!\!\!\!/\;$ does not hold. If $P_1 \xrightarrow{\mathsf{tick}} \pi_1$ does not hold, then by the inductive hypothesis we have $P_1 \xrightarrow{\lambda} \pi_1$ for some $\lambda \in \{\tau, a!v, s?(x)\}$, and by rule (Par) we infer $P_1 \parallel P_2 \xrightarrow{\lambda} \pi_1 \parallel \overline{P_2}$, which gives the thesis. If $P_2 \xrightarrow{\mathsf{tick}} \pi_2$ does not hold, then by the inductive hypothesis we have $P_2 \xrightarrow{\lambda} \pi_2$ for some $\lambda \in \{\tau, a!v, s?(x)\}$, and by the rule symmetric to (Par) we infer $P_1 \parallel P_2 \xrightarrow{\lambda} \overline{P_1} \parallel \pi_2$, which gives the thesis. If $P_1 \parallel P_2 \xrightarrow{\tau}\!\!\!\!\!/\;$ does not hold then there is some transition $P_1 \parallel P_2 \xrightarrow{\tau} \pi$, which gives the thesis. The cases $P = [b]\{P_1\},\{P_2\}$ and $P = Q\backslash c$ are similar.

(d) The well-timedness property is straightforward from time–guardedness recursion.

□

The challenge in the proof of Theorem 2.9 is to lift the results of Lemma Appendix A.1 to CPSs.

Proof of Theorem 2.9

(a) We note that transitions labelled tick can be derived only by rule (Time). Therefore, from the hypothesis $M \xrightarrow{\mathsf{tick}} \gamma$ and $M \xrightarrow{\mathsf{tick}} \gamma'$ with $M = E; S \bowtie P$, we infer that there are process distributions π and π' such that

$$\frac{P \xrightarrow{\mathsf{tick}} \pi \qquad S \bowtie P \xrightarrow{\tau}\!\!\!\!\!/\; \qquad S \in inv}{S \bowtie P \xrightarrow{\mathsf{tick}} next_E(S) \bowtie \pi} \quad \text{and} \quad \frac{P \xrightarrow{\mathsf{tick}} \pi' \qquad S \bowtie P \xrightarrow{\tau}\!\!\!\!\!/\; \qquad S \in inv}{S \bowtie P \xrightarrow{\mathsf{tick}} next_E(S) \bowtie \pi'}$$

where $\gamma = E; next_E(S) \bowtie \pi$ and $\gamma' = E; next_E(S) \bowtie \pi'$. By the property of time determinism for processes in Lemma Appendix A.1 we infer that $P \xrightarrow{\mathsf{tick}} \pi$ and $P \xrightarrow{\mathsf{tick}} \pi'$ imply $\pi \equiv \pi'$, hence $\gamma \equiv \gamma'$, which completes the proof.

(b) From the hypothesis $M \xrightarrow{\tau} \gamma$ with $M = E; S \bowtie P$, we infer that $\gamma = E; \sigma \bowtie \pi$ for distributions σ and π such that $S \bowtie P \xrightarrow{\tau} \sigma \bowtie \pi$ is derived from the rules in Table 2. To show the thesis that no transition from M labelled tick can be derived, it is enough to show that no transition from $S \bowtie P$ labelled tick can be derived from the rules in Table 2. This follows by the fact that the only rule which may be applied to infer any tick-labelled transition from $S \bowtie P$ is rule (Time), which cannot be applied since it has $S \bowtie P \xrightarrow{\tau}\!\!\!\!/$ among its premises.

(c) From the hypothesis that $M \xrightarrow{\mathsf{tick}} \gamma$ with $M = E; S \bowtie P$ cannot be inferred for any distribution γ, we infer that $S \bowtie P \xrightarrow{\mathsf{tick}} \sigma \bowtie \pi$ cannot be derived for any σ and π from the rules in Table 2. Therefore, at least one of the premises $P \xrightarrow{\mathsf{tick}} \pi$, $S \bowtie P \xrightarrow{\tau}\!\!\!\!/$ and $S \in inv$ of rule (Time) does not hold. If premise $P \xrightarrow{\mathsf{tick}} \pi$ does not hold for any π, then by the property of patience for processes in Lemma Appendix A.1 we have $P \xrightarrow{\lambda} \pi'$ for some π' and $\lambda \in \{\tau, a!v, s?(x)\}$. Let us consider the case $\lambda = \tau$. From $P \xrightarrow{\tau} \pi'$, either $S \in inv$ is not valid, or we can apply rule (Tau) to infer the transition $S \bowtie P \xrightarrow{\tau} \overline{S} \bowtie \pi'$, which gives $M \xrightarrow{\tau} E; \overline{S} \bowtie \pi'$. In both cases the thesis holds. The cases $\lambda \in \{a!v, s?(x)\}$ can be proved similarly by using rules (ActWrite) and (SensRead), respectively. If premise $P \xrightarrow{\mathsf{tick}} \pi$ holds for some π then either premises $S \in inv$ or premise $S \bowtie P \xrightarrow{\tau}\!\!\!\!/$ does not hold. In the former case the thesis follows. In the latter case we have a τ-labelled transition from M and the thesis holds as well.

(d) The proof is by contradiction. Suppose there is no k satisfying the statement of the thesis. Hence there exists an unbounded derivation

$$E; S \bowtie P = E; S_1 \bowtie P_1 \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_n} E; S_n \bowtie P_n \xrightarrow{\alpha_{n+1}} \ldots$$

with $\alpha_i \neq \mathsf{tick}$ for $i \geq 1$, namely there exist distributions $\sigma_i \bowtie \pi_i$ for $i \geq 1$ with $\sigma_i \bowtie P_i \xrightarrow{\alpha_i} \sigma_{i+1} \bowtie \pi_{i+1}$, $S_{i+1} \in \mathsf{supp}(\sigma_{i+1})$ and $P_{i+1} \in \mathsf{supp}(\pi_{i+1})$. This contradicts the property of well-timedness for processes in Lemma Appendix A.1. □

Appendix A.2. Proofs of Section 4

In order to prove Proposition 4.1 and Proposition 4.2 we use the following lemma that formalises the invariant properties binding the state variable *temp* with the activity of the cooling system. Intuitively, when the cooling system is inactive then the value of the state variable *temp* lays in the interval $[0, 11 + err + \delta]$. Furthermore, if the coolant is not active and the variable *temp* lays in the interval $(10 + err, 11 + err + \delta]$ then the cooling will be turned on in the next time slot. Finally, if the cooling system is active then there is some $k = 1 \ldots 5$ such that the system was activated k time units ago, it was kept active so far and the state variable *temp* lays in the real interval $(10 - err - k{*}(1{+}\delta), 11 + err + \delta - k{*}(1{-}\delta)]$.

Lemma Appendix A.2. *Let Eng_g be the system defined in Section 4. Let us consider an arbitrace execution trace of Eng_g of the form*

$$Eng_g = M_1 \xrightarrow{t_1} \xrightarrow{\mathsf{tick}} M_2 \xrightarrow{t_2} \xrightarrow{\mathsf{tick}} \cdots \xrightarrow{t_{n-1}} \xrightarrow{\mathsf{tick}} M_n$$

where the sub-traces t_j contain no tick*-actions, for any $j \in 1 \ldots n{-}1$, and for any $i \in 1 \ldots n$ we have $M_i = Env_g; S_i \bowtie P_i$ with $S_i = \langle \xi^i_{\mathrm{x}}, \xi^i_{\mathrm{s}}, \xi^i_{\mathrm{a}} \rangle$ and $Env_g = \langle evol, meas, inv \rangle$. Then, for any $i \in 1 \ldots n{-}1$ we have the following:*

1. *if $\xi^i_{\mathrm{a}}(cool) = \mathsf{off}$ then $\xi^i_{\mathrm{x}}(temp) \in [0, 11 + err + \delta]$;*
2. *if $\xi^i_{\mathrm{a}}(cool) = \mathsf{off}$ and $\xi^i_{\mathrm{x}}(temp) \in (10 + err, 11 + err + \delta]$ then, in the next time slot, $\xi^{i+1}_{\mathrm{a}}(cool) = \mathsf{on}$;*
3. *if $\xi^i_{\mathrm{a}}(cool) = \mathsf{on}$ then $\xi^i_{\mathrm{x}}(temp) \in (10 - err - k{*}(1{+}\delta), 11 + err + \delta - k{*}(1{-}\delta)]$, for some $k \in 1 \ldots 5$ such that $\xi^{i-k}_{\mathrm{a}}(cool) = \mathsf{off}$ and $\xi^{i-j}_{\mathrm{a}}(cool) = \mathsf{on}$, for all $j \in 0 \ldots k{-}1$.*

Proof. Let us denote with v_i the values of the state variable *temp* in the systems M_i, i.e., $\xi^i_{\mathrm{x}}(temp) = v_i$. Moreover we will say that the coolant is active (resp., is not active) in M_i if $\xi^i_{\mathrm{a}}(cool) = \mathsf{on}$ (resp., $\xi^i_{\mathrm{a}}(cool) = \mathsf{off}$).

The proof is by mathematical induction on n, i.e., the number of tick-actions of our traces.

The case base $n = 1$ follows directly from the definition of Eng_g. Let prove the inductive case. We assume that the three statements holds for $n - 1$ and we prove that they also hold for n.

1. Let us assume that the cooling is not active in M_n, then we prove that $v_n \in [0, 11 + err + \delta]$. We consider separately the cases in which the coolant is active or not in M_{n-1}.
   - Suppose the coolant is not active in M_{n-1} (and inactive in M_n).
     By the inductive hypothesis we have $v_{n-1} \in [0, 11 + err + \delta]$. Since we know that in M_n the cooling is not active, it follows that $v_{n-1} \in [0, 10 + err]$, the reason being that $v_{n-1} \in (10 + err, 11 + \epsilon + \delta]$ and the inductive hypothesis would imply that the coolant is active in M_n. Furthermore, in M_n the temperature will increase of a value laying in the interval $[1-\delta, 1+\delta]_g = [0.6, 1.4]_g$. Thus v_n will be in $[0.6, 11+err+\delta] \subseteq [0, 11+err+\delta]$.
   - Suppose the coolant is active in M_{n-1} (and inactive in M_n).
     By the inductive hypothesis we have $v_{n-1} \in (10 - err - k * (1 + \delta), 11 + err + \delta - k * (1 - \delta)]$ for some $k \in 1 \dots 5$ such that the coolant is not active in M_{n-1-k} and is active in all $M_{n-k}, \dots, M_{n-1}$.
     The case $k \in \{1, \dots, 4\}$ is not admissible, the reason being that $k \in \{1, \dots, 4\}$ together with the fact that the coolant is inactive in M_n would imply that the coolant bas been kept active for less than 5 steps, which cannot happen.
     Hence it must be $k = 5$. Since $\delta = 0.4$, $err = 0.1$ and $k = 5$, it holds that $v_{n-1} \in (10 - 0.1 - 5 * 1.4, 11 + 0.1 + 0.4 - 5 * 0.6] = (2.8, 8.6]$. Moreover, since the coolant is active for 5 tick actions, the controller of M_{n-1} checks the temperature. However, since $v_{n-1} \in (2.8, 8.6]$ then the coolant is turned off. Thus, in the next time slot, the temperature will increase of a value in $[1 - \delta, 1 + \delta]_g = [0.6, 1.4]_g$. As a consequence in M_n we will have $v_n \in [2.8 + 0, 6, 8.6 + 1.4] = [3.4, 10] \subseteq [0, 11 + err + \delta]$.
2. Let us assume that the coolant is not active in M_n and $v_n \in (10 + err, 11 + err + \delta]$, then we prove that the coolant is active in M_{n+1}. Since the coolant is not active in M_n then it will check the temperature before the next time slot. Since $v_n \in (10 + err, 11 + err + \delta]$ and $err = 0.1$, then the process *Ctrl* will sense a temperature greater than 10 and the coolant will be turned on. Thus the coolant will be active in M_{n+1}.
3. Let us assume that the coolant is active in M_n, then we prove that $v_n \in (10-err-k*(1+\delta), 11+err+\delta-k*(1-\delta)]$ for some $k \in 1 \dots 5$ and the coolant is not active in M_{n-k} and active in all $M_{n-k+1}, \dots, M_n$.
   We separate the case in which the coolant is active in M_{n-1} from that in which is not active.
   - Suppose the coolant is not active in M_{n-1} (and active in M_n).
     In this case $k = 1$ as the coolant is not active in M_{n-1} and it is active in M_n. Since $k = 1$, we have to prove $v_n \in (10 - err - (1 + \delta), 11 + err + \delta - (1 - \delta)]$.
     However, since the coolant is not active in M_{n-1} and is active in M_n it means that the coolant has been switched on in M_{n-1} because the sensed temperature was above 10 (this may happen only if $v_{n-1} > 10 - err$). By inductive hypothesis, since the coolant is not active in M_{n-1}, we have that $v_{n-1} \in [0, 11 + err + \delta]$. Therefore, from $v_{n-1} > 10 - err$ and $v_{n-1} \in [0, 11 + err + \delta]$ it follows that $v_{n-1} \in (10 - err, 11 + err + \delta]$. Furthermore, since the coolant is active in M_n, the temperature will decrease of a value in $[1 - \delta, 1 + \delta]_g$ and therefore $v_n \in (10 - err - (1 + \delta), 11 + err + \delta - (1 - \delta)]$ which concludes this case of the proof.
   - Suppose the coolant is active in M_{n-1} (and active in M_n as well).
     By inductive hypothesis there is $h \in 1 \dots 5$ such that $v_{n-1} \in (10 - err - h * (1 + \delta), 11 + err + \delta - h * (1 - \delta)]$ and the coolant is not active in M_{n-1-h} and is active in $M_{n-h}, \dots, M_{n-1}$.
     The case $h = 5$ is not admissible. In fact, since $\delta = 0.4$ and $err = 0.1$, if $h = 5$ then $v_{n-1} \in (10 - 0.1 - 5 * 1.4, 11 + 0.1 + \delta - 5 * 0.6] = (2.8, 8.6]$. Furthermore, since the coolant is already active since 5 tick actions, the controller of M_{n-1} is supposed to check the temperature. As $v_{n-1} \in (2.8, 8.6]$ the coolant should be turned off. In contradiction with the the fact that the coolant is active in M_n.
     Hence it must be $h \in 1 \dots 4$. Let us prove that for $k = h + 1$ we obtain our result. Namely we have to prove that, for $k = h + 1$, (i) $v_n \in (10 - err - k * (1 + \delta), 11 + err + \delta - k * (1 - \delta)]$, and (ii) the coolant is not active in M_{n-k} and active in all $M_{n-k+1}, \dots, M_n$.
     Let us prove the statement (i). By inductive hypotheses, it holds that $v_{n-1} \in (10 - err - h * (1 + \delta), 11 + err + \delta - h * (1 - \delta)]$. Since the coolant is active in M_n then the temperature will decrease. Hence, $v_n \in (10 - err - (h + 1) * (1 + \delta), 11 + err + \delta - (h + 1) * (1 - \delta)]$. Therefore, since $k = h + 1$, we have that $v_n \in (10 - err - k * (1 + \delta), 11 + err + \delta - k * (1 - \delta)]$.
     Let us prove the statement (ii). By inductive hypothesis the coolant is inactive in M_{n-1-h} and it is active in all $M_{n-h}, \dots, M_{n-1}$. Now, since the coolant is active in M_n, for $k = h + 1$, we have that the coolant is not active in M_{n-k} and is active in all $M_{n-k+1}, \dots, M_n$ which concludes this case of the proof.

□

Proof of Proposition 4.1 By the first two items of Lemma Appendix A.2 and since $\delta = 0.4$ and $err = 0.1$, we infer that the value of the state variable *temp* is always in the real interval $[0, 11.5]$. As a consequence, the invariant of the system is never violated and the system never deadlocks. Then, the last item of Lemma Appendix A.2 ensures that after 5 tick-actions happening when the coolant is active, the state variable *temp* is always in the real interval $(10 - 0.1 - 5 * 1.4, 11 + 0.1 + 0.4 - 5 * 0.6] = (2.9, 8.5]$. Hence the process *Ctrl* will never transmit on the channel *warning*. □

Proof of Proposition 4.2 Let us prove the two statements separately.

- If process *Ctrl* senses a temperature above 10 (and hence *Eng* turns on the cooling) then the value of the state variable *temp* is greater than $10 - err$. By Lemma Appendix A.2 the value of the state variable *temp* is always less or equal than $11 + err + \delta$. Therefore, if *Ctrl* senses a temperature above 10, then the value of the state variable *temp* is in $(10 - err, 11 + err + \delta] = (9.9, 11.5]$.
- By Lemma Appendix A.2 (third item) the coolant can be active for no more than 5 time slots. Hence, by Lemma Appendix A.2, when *Eng* turns off the cooling system the state variable *temp* ranges over $(10 - err - 5 * (1 + \delta), 11 + err + \delta - 5 * (1 - \delta)] = (2.9, 8.5]$.

□

Proof of Proposition 4.4 It is is enough to prove that there exists an execution trace of the engine $\widehat{Eng}_g$ containing an output along channel *warning*. Then the result follows by an application of Proposition 4.1.

We prove the thesis for $g = 1$. Indeed a trace of $\widehat{Eng}_g$ with $g = 1$ is a trace of $\widehat{Eng}_{g'}$ with $g' \geq g$.

We can easily build up a trace for $\widehat{Eng}_g$ with $g = 1$ in which, after 10 tick-actions, in the 11-th time slot, the value of the state variable *temp* is 10.1. In fact, it is enough to increase the temperature of 1 degree for the first 9 rounds and an increase of 1.1 degrees in the 10-th time slot. Notice that these are admissible values, since both 1 and 1.1 are in $[1 - \delta, 1 + \delta]_g = [0.6, 1.4]_g$ with $g = 1$. Being 10.1 the value of the state variable *temp*, there is an execution trace in which the sensed temperature is 10 (recall that $err = 0.1$ and $-0.1 \in [-0.1, 0.1]_g$ with $g = 1$) and hence the cooling system is not activated. However, in the following time slot, i.e. the 12-th time slot, the temperature may reach the value $10.1 + 1 + \delta = 11.5$, imposing the activation of the cooling system. After 5 time units of cooling, in the 17-th time slot, the variable *temp* could be $11.5 - 5 * (0.7 - \delta) = 11.5 - 1.5 = 10$. The sensed temperature would be in the real interval $[9.9, 10.1]_g$ with $g = 1$. Thus, there is an execution trace in which the sensed temperature is 10.1. As a consequence, the warning will be emitted, in the 17-th time slot. □

Appendix A.3. Proofs of Section 5

To prove that all $\mathbf{d}^n$ are 1-bounded pseudometrics (Proposition 5.8), we need some preliminary results. First we show that the Kantorovich functional $\mathbf{K}$ maps pseudometrics to pseudometrics.

Proposition Appendix A.3. *If* $d\colon \mathcal{pCCPS} \times \mathcal{pCCPS} \to [0,1]$ *is a 1-bounded pseudometric, then also* $\mathbf{K}(d)\colon \mathcal{D}(\mathcal{pCCPS}) \times \mathcal{D}(\mathcal{pCCPS})$ *is a 1-bounded pseudometric.*

Proof. To show $\mathbf{K}(d)(\gamma, \gamma) = 0$ for all $\gamma \in \mathcal{D}(\mathrm{pCCPS})$ it is enough to take the matching $\omega \in \Omega(\gamma, \gamma)$ defined by $\omega(M, M) = \gamma(M)$, for all $M \in \mathrm{pCCPS}$, and $\omega(M, N) = 0$, for all $M, N \in \mathrm{pCCPS}$ with $M \neq N$. In fact, we have $\mathbf{K}(d)(\gamma, \gamma) \leq \sum_{M,N \in \mathrm{pCCPS}} \omega(M, N) \cdot d(M, N) = \sum_{M \in \mathrm{pCCPS}} \gamma(M) \cdot d(M, M) = 0$.

The symmetry $\mathbf{K}(d)(\gamma, \gamma') = \mathbf{K}(d)(\gamma', \gamma)$ for all $\gamma, \gamma' \in \mathcal{D}(\mathrm{pCCPS})$ follows directly by the fact that if we take two functions $\omega, \omega'\colon \mathrm{pCCPS} \times \mathrm{pCCPS} \to [0, 1]$ such that $\omega(M, N) = \omega'(N, M)$ for all $M, N \in \mathrm{pCCPS}$, then $\omega \in \Omega(\gamma, \gamma')$ if and only if $\omega' \in \Omega(\gamma', \gamma)$.

It remains to prove the triangle inequality $\mathbf{K}(d)(\gamma_1, \gamma_2) \leq \mathbf{K}(d)(\gamma_1, \gamma_3) + \mathbf{K}(d)(\gamma_3, \gamma_2)$ for all $\gamma_1, \gamma_2, \gamma_3 \in \mathcal{D}(\mathrm{pCCPS})$. First we consider the function $\omega\colon \mathrm{pCCPS} \times \mathrm{pCCPS} \to [0, 1]$ defined for all $M_1, M_2 \in \mathrm{pCCPS}$ as $\omega(M_1, M_2) = \sum_{M_3 \in \mathrm{pCCPS} | \gamma_3(M_3) \neq 0} \frac{\omega_1(M_1, M_3) \cdot \omega_2(M_3, M_2)}{\gamma_3(M_3)}$, where the function $\omega_1 \in \Omega(\gamma_1, \gamma_3)$ is one of the optimal matchings realising $\mathbf{K}(d)(\gamma_1, \gamma_3)$ and $\omega_2 \in \Omega(\gamma_3, \gamma_2)$ one of the optimal matchings realising $\mathbf{K}(d)(\gamma_3, \gamma_2)$. Then, we prove that (i) ω is a

matching in $\Omega(\gamma_1,\gamma_2)$, and (ii) $\sum_{M_1,M_2\in \mathtt{pCCPS}} \omega(M_1,M_2)\cdot d(M_1,M_2) \leq \mathbf{K}(d)(\gamma_1,\gamma_3)+\mathbf{K}(d)(\gamma_3,\gamma_2)$, which immediately implies $\mathbf{K}(d)(\gamma_1,\gamma_2)\leq \mathbf{K}(d)(\gamma_1,\gamma_3)+\mathbf{K}(d)(\gamma_3,\gamma_2)$. To show (i) we prove that the left marginal of ω is γ_1 by

$$\begin{array}{lll}
& \sum_{M_2\in \mathtt{pCCPS}} \omega(M_1,M_2) & \\
= & \sum_{M_2\in \mathtt{pCCPS}} \sum_{M_3\in \mathtt{pCCPS}|\gamma_3(M_3)\neq 0} \frac{\omega_1(M_1,M_3)\cdot\omega_2(M_3,M_2)}{\gamma_3(M_3)} & \\
= & \sum_{M_3\in \mathtt{pCCPS}|\gamma_3(M_3)\neq 0} \frac{\omega_1(M_1,M_3)\cdot\gamma_3(M_3)}{\gamma_3(M_3)} & (\text{by } \omega_2\in\Omega(\gamma_3,\gamma_2)) \\
= & \sum_{M_3\in \mathtt{pCCPS}|\gamma_3(M_3)\neq 0} \omega_1(M_1,M_3) & \\
= & \gamma_1(M_1) & (\text{by } \omega_1\in\Omega(\gamma_1,\gamma_3))
\end{array}$$

and we observe that the proof that the right marginal of ω is γ_2 is analogous. Then, we show (ii) by

$$\begin{array}{lll}
& \sum_{M_1,M_2\in \mathtt{pCCPS}} \omega(M_1,M_2)\cdot d(M_1,M_2) & \\
= & \sum_{M_1,M_2\in \mathtt{pCCPS}} \sum_{M_3\in \mathtt{pCCPS}|\gamma_3(M_3)\neq 0} \frac{\omega_1(M_1,M_3)\cdot\omega_2(M_3,M_2)}{\gamma_3(M_3)}\cdot d(M_1,M_2) & \\
\leq & \sum_{M_1,M_2\in \mathtt{pCCPS}, M_3\in \mathtt{pCCPS}|\gamma_3(M_3)\neq 0} \frac{\omega_1(M_1,M_3)\cdot\omega_2(M_3,M_2)}{\gamma_3(M_3)}\cdot d(M_1,M_3) + & \\
& \sum_{M_1,M_2\in \mathtt{pCCPS}, M_3\in \mathtt{pCCPS}|\gamma_3(M_3)\neq 0} \frac{\omega_1(M_1,M_3)\cdot\omega_2(M_3,M_2)}{\gamma_3(M_3)}\cdot d(M_3,M_2) & \\
= & \sum_{M_1,M_3\in \mathtt{pCCPS}} \frac{\omega_1(M_1,M_3)\cdot\gamma_3(M_3)}{\gamma_3(M_3)}\cdot d(M_1,M_3) + \sum_{M_2,M_3\in \mathtt{pCCPS}} \frac{\gamma_3(M_3)\cdot\omega_2(M_3,M_2)}{\gamma_3(M_3)}\cdot d(M_3,M_2) & \\
= & \sum_{M_1,M_3\in \mathtt{pCCPS}} \omega_1(M_1,M_3)\cdot d(M_1,M_3) + \sum_{M_2,M_3\in \mathtt{pCCPS}} \omega_2(M_3,M_2)\cdot d(M_3,M_2) & \\
= & \mathbf{K}(d)(\gamma_1,\gamma_3)+\mathbf{K}(d)(\gamma_3,\gamma_2) &
\end{array}$$

where the inequality follows from the triangular property of d and the third last equality follows by $\omega_2\in\Omega(\gamma_3,\gamma_2)$ and $\omega_1\in\Omega(\gamma_1,\gamma_3)$. □

Now we show that, given any weak bisimulation metric d with $d(M,N)<1$, then N can mimic weak transitions $M \stackrel{\hat{\alpha}}{\Longrightarrow}$ besides those of the form $M \stackrel{\alpha}{\longrightarrow}$.

Lemma Appendix A.4. *Assume a weak bisimulation metric d and $M,N\in \mathtt{pCCPS}$ with $d(M,N)<1$. If $M \stackrel{\hat{\alpha}}{\Longrightarrow} \gamma_M$, then there is a transition $N \stackrel{\hat{\alpha}}{\Longrightarrow} \gamma_N$ such that $\mathbf{K}(d)(\gamma_M+(1-|\gamma_M|)\overline{\mathsf{Dead}},\gamma_N+(1-|\gamma_N|)\overline{\mathsf{Dead}})\leq d(M,N)$.*

Proof. We proceed by induction on the length n of the weak transition $M \stackrel{\hat{\alpha}}{\Longrightarrow} \gamma_M$.

Base case $n=1$. We have two sub-cases: The first is $\alpha=\tau$ and $\gamma_M=\overline{M}$, the second is $M \stackrel{\alpha}{\longrightarrow} \gamma_M$. In the first case, by the definition of $\stackrel{\widehat{\tau}}{\Longrightarrow}$ we have that $N \stackrel{\widehat{\tau}}{\Longrightarrow} \overline{N}$ and, then, the thesis holds for $\gamma_N=\overline{N}$ by observing that $\mathbf{K}(d)(\overline{M}+(1-|\overline{M}|)\overline{\mathsf{Dead}},\overline{N}+(1-|\overline{N}|)\overline{\mathsf{Dead}})=\mathbf{K}(d)(\overline{M},\overline{N})=d(M,N)$. In the second case, the thesis follows directly by the definition of weak simulation metric.

Inductive case $n>1$. The derivation $M \stackrel{\hat{\alpha}}{\Longrightarrow} \gamma_M$ is obtained by $M \stackrel{\hat{\beta}_1}{\Longrightarrow} \rho_M$ and $\rho_M \stackrel{\hat{\beta}_2}{\longrightarrow} \gamma_M$, for some distribution $\rho_M\in\mathcal{D}(\mathtt{pCCPS})$. The length of the derivation $M \stackrel{\hat{\beta}_1}{\Longrightarrow} \rho_M$ is $n-1$ and hence, by the inductive hypothesis, there is a transition $N \stackrel{\hat{\beta}_1}{\Longrightarrow} \rho_N$ such that $\mathbf{K}(d)(\rho_M+(1-|\rho_M|)\overline{\mathsf{Dead}},\rho_N+(1-|\rho_N|)\overline{\mathsf{Dead}})\leq d(M,N)$. The sub-distributions ρ_M and ρ_N are of the form $\rho_M=\sum_{i\in I} p_i\cdot\overline{M_i}$ and $\rho_N=\sum_{j\in J} q_j\cdot\overline{N_j}$. We have two sub-cases: The first is $\beta_1=\tau$ and $\beta_2=\alpha$, the other $\beta_1=\alpha$ and $\beta_2=\tau$.

We consider the case $\beta_1=\tau$ and $\beta_2=\alpha$, the other is analogous. In this case we have $|\rho_M|=|\rho_N|=1$ and $\mathbf{K}(d)(\rho_M,\rho_N)\leq d(M,N)$. The transition $\rho_M \stackrel{\hat{\beta}_2}{\longrightarrow} \gamma_M$ is derived from a β_2-transition by some of the CPSs M_i, namely I is partitioned into sets $I_1\cup I_2$ such that for all $i\in I_1$ we have $M_i \stackrel{\beta_2}{\longrightarrow} \gamma_i$ for suitable distributions γ_i, for each $i\in I_2$ we have $M_i \stackrel{\beta_2}{\not\longrightarrow}$, and $\rho_M=\sum_{i\in I_1} p_i\cdot\gamma_i$. Analogously, J is partitioned into sets $J_1\cup J_2$ such that for all $j\in J_1$ we have $N_j \stackrel{\hat{\beta}_2}{\Longrightarrow} \gamma_j$ for suitable distributions γ_j and for each $j\in J_2$ we have $N_j \stackrel{\hat{\beta}_2}{\not\Longrightarrow}$. This gives $\rho_N \stackrel{\hat{\beta}_2}{\Longrightarrow} \gamma_N$ with $\gamma_N=\sum_{j\in J_1} q_j\cdot\gamma_j$. Since

we had $N \overset{\hat{\beta}_1}{\Longrightarrow} \rho_N$, we can conclude $N \overset{\hat{\alpha}}{\Longrightarrow} \gamma_N$. In the following we prove that the transitions $N_j \overset{\hat{\beta}_2}{\Longrightarrow} \gamma_j$ can be chosen so that $\mathbf{K}(d)(\gamma_M + (1- |\gamma_M|)\mathsf{Dead}, \gamma_N + (1- |\gamma_N|)\mathsf{Dead}) \leq d(M,N)$, which concludes the proof.

Let ω be one of the optimal matchings realising $\mathbf{K}(d)(\rho_M, \rho_N)$. We can rewrite the distributions ρ_M and ρ_N as $\rho_M = \sum_{i\in I, j\in J} \omega(M_i, N_j) \cdot \overline{M_i}$ and $\rho_N = \sum_{i\in I, j\in J} \omega(M_i, N_j) \cdot \overline{N_j}$. For all $i \in I_1$ and $j \in J$, define $\gamma_{i,j} = \gamma_i$. We can rewrite γ_M as $\gamma_M = \sum_{i\in I_1, j\in J} \omega(M_i, N_j) \cdot \gamma_{i,j}$. Analogously, for each $j \in J_1$ and $i \in I$ we note that the transition $q_j \cdot \overline{N_j} \overset{\hat{\beta}_2}{\Longrightarrow} \gamma_j$ can always be split into $\sum_{i\in I} \omega(M_i, N_j) \cdot \overline{N_j} \overset{\hat{\beta}_2}{\Longrightarrow} \sum_{i\in I} \omega(M_i, N_j) \cdot \gamma'_{i,j}$ so that we can rewrite γ_j as $\gamma_j = \sum_{i\in I} \omega(M_i, N_j) \cdot \gamma'_{i,j}$ and γ_N as $\gamma_N = \sum_{i\in I, j\in J_1} \omega(M_i, N_j) \cdot \gamma'_{i,j}$. Then we note that for all $i \in I_1$ and $j \in J_1$ with $d(M_i, N_j) < 1$, the transition $N_j \overset{\hat{\beta}_2}{\Longrightarrow} \gamma'_{i,j}$ can be chosen so that $\mathbf{K}(d)(\gamma_{i,j}, \gamma'_{i,j} + (1- |\gamma'_{i,j}|)\mathsf{Dead}) \leq d(M_i, N_j)$.

For all $i \in I_1$ and $j \in J_1$ with $d(M_i, N_j) < 1$, let us assume that $\omega_{i,j}$ is one of the optimal matchings realising $\mathbf{K}(d)(\gamma_{i,j}, \gamma_j + (1- |\gamma_j|)\mathsf{Dead})$. Define $\omega' \colon \mathtt{pCCPS} \times \mathtt{pCCPS} \to [0,1]$ as the function such that

$$\omega'(M', N') = \begin{cases} \sum_{i\in I_1, j\in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(M', N') & \text{if } M' \neq \mathsf{Dead} \neq N' \\ \sum_{i\in I_1, j\in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(M', N') + \sum_{i\in I_1, j\in J_2} \omega(M_i, N_j) \cdot \gamma_{i,j}(M') & \text{if } M' \neq \mathsf{Dead} = N' \\ \sum_{i\in I_1, j\in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(M', N') + \sum_{i\in I_2, j\in J_1} \omega(M_i, N_j) \cdot \gamma'_{i,j}(N') & \text{if } M' = \mathsf{Dead} \neq N' \\ \sum_{i\in I_1, j\in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(M', N') + \sum_{i\in I_1, j\in J_2} \omega(M_i, N_j) \cdot \gamma_{i,j}(M') & \\ + \sum_{i\in I_2, j\in J_1} \omega(M_i, N_j) \cdot \gamma'_{i,j}(N') + \sum_{i\in I_2, j\in J_2} \omega(M_i, N_j) & \text{if } M' = \mathsf{Dead} = N'. \end{cases}$$

To infer the proof obligation $\mathbf{K}(d)(\gamma_M + (1- |\gamma_M|)\overline{\mathsf{Dead}}, \gamma_N + (1- |\gamma_N|)\overline{\mathsf{Dead}}) \leq d(M,N)$ we show that (i) ω' is a matching in $\Omega(\gamma_M + (1- |\gamma_M|)\overline{\mathsf{Dead}}, \gamma_N + (1- |\gamma_N|)\overline{\mathsf{Dead}})$, and (ii) $\sum_{M',N'\in \mathtt{pCCPS}} \omega'(M', N') \cdot d(M', N') \leq d(M,N)$.

To show (i) we prove that the left marginal of ω' is $\gamma_M + (1- |\gamma_M|)\overline{\mathsf{Dead}}$. The proof that the right marginal is $\gamma_N + (1- |\gamma_N|)\overline{\mathsf{Dead}})$ is analogous. For any CPS $M' \neq \mathsf{Dead}$, we have

$$\begin{aligned} & \textstyle\sum_{N'\in \mathtt{pCCPS}} \omega'(M', N') \\ = \; & \textstyle\sum_{N'\neq \mathsf{Dead}} \sum_{i\in I_1, j\in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(M', N') + \sum_{i\in I_1, j\in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(M', \mathsf{Dead}) + \sum_{i\in I_1, j\in J_2} \omega(M_i, N_j) \cdot \gamma_{i,j}(M') \\ = \; & \textstyle\sum_{i\in I_1, j\in J_1} \omega(M_i, N_j) \sum_{N'\in \mathtt{pCCPS}} \omega_{i,j}(M', N') + \sum_{i\in I_1, j\in J_2} \omega(M_i, N_j) \cdot \gamma_{i,j}(M') \\ = \; & \textstyle\sum_{i\in I_1, j\in J_1} \omega(M_i, N_j) \cdot \gamma_{i,j}(M') + \sum_{i\in I_1, j\in J_2} \omega(M_i, N_j) \cdot \gamma_{i,j}(M') \\ = \; & \textstyle\sum_{i\in I_1, j\in J} \omega(M_i, N_j) \cdot \gamma_{i,j}(M') \\ = \; & (\gamma_M + (1- |\gamma_M|)\mathsf{Dead})(M') \end{aligned}$$

with the third equality by the fact that $\omega_{i,j}$ is a matching in $\Omega(\gamma_{i,j}, \gamma'_{i,j})$.

Consider now the CPS Dead. In this case we have that

$$\begin{aligned} & \textstyle\sum_{N'\in \mathtt{pCCPS}} \omega'(\mathsf{Dead}, N') \\ = \; & \textstyle\sum_{N'\neq \mathsf{Dead}} \sum_{i\in I_1, j\in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(\mathsf{Dead}, N') + \sum_{N'\neq \mathsf{Dead}} \sum_{i\in I_2, j\in J_1} \omega(M_i, N_j) \cdot \gamma'_{i,j}(N') \\ & \textstyle + \sum_{i\in I_1, j\in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(\mathsf{Dead}, \mathsf{Dead}) + \sum_{i\in I_1, j\in J_2} \omega(M_i, N_j) \cdot \gamma_{i,j}(\mathsf{Dead}) \\ & \textstyle + \sum_{i\in I_2, j\in J_1} \omega(M_i, N_j) \cdot \gamma'_{i,j}(\mathsf{Dead}) + \sum_{i\in I_2, j\in J_2} \omega(M_i, N_j) \\ = \; & \textstyle\sum_{N'\in \mathtt{pCCPS}} \sum_{i\in I_1, j\in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(\mathsf{Dead}, N') + \sum_{N'\in \mathtt{pCCPS}} \sum_{i\in I_2, j\in J_1} \omega(M_i, N_j) \cdot \gamma'_{i,j}(N') \\ & \textstyle + \sum_{i\in I_1, j\in J_2} \omega(M_i, N_j) \cdot \gamma_{i,j}(\mathsf{Dead}) + \sum_{i\in I_2, j\in J_2} \omega(M_i, N_j) \\ = \; & \textstyle\sum_{i\in I_1, j\in J_1} \omega(M_i, N_j) \cdot \gamma_{i,j}(\mathsf{Dead}) + \sum_{i\in I_2, j\in J_1} \omega(M_i, N_j) \\ & \textstyle + \sum_{i\in I_1, j\in J_2} \omega(M_i, N_j) \cdot \gamma_{i,j}(\mathsf{Dead}) + \sum_{i\in I_2, j\in J_2} \omega(M_i, N_j) \\ = \; & \textstyle\sum_{i\in I_1, j\in J} \omega(M_i, N_j) \cdot \gamma_{i,j}(\mathsf{Dead}) + \sum_{i\in I_2, j\in J} \omega(M_i, N_j) \\ = \; & (\gamma_M + (1- |\gamma_M|)\overline{\mathsf{Dead}})(\mathsf{Dead}) \end{aligned}$$

where the third equality follows by observing that, being the function$\omega_{i,j}$ a matching in $\Omega(\gamma_{i,j}, \gamma'_{i,j})$, then we have that $\sum_{N' \in \mathtt{pCCPS}} \sum_{i \in I_1, j \in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(\mathsf{Dead}, N') = \sum_{i \in I_1, j \in J_1} \omega(M_i, N_j) \cdot \gamma_{i,j}(\mathsf{Dead})$, and being $\gamma'_{i,j}$ a distribution, then $\sum_{N' \in \mathtt{pCCPS}} \sum_{i \in I_2, j \in J_1} \omega(M_i, N_j) \cdot \gamma'_{i,j}(N') = \sum_{i \in I_2, j \in J_1} \omega(M_i, N_j)$, and the last equality by $\sum_{i \in I_1, j \in J} \omega(M_i, N_j) = \sum_{i \in I_1} p_i = |\gamma_M|$.

To prove (ii), by looking at the definition of ω' above we get that $\sum_{M', N' \in \mathtt{pCCPS}} \omega'(M', N') \cdot d(M', N')$ is the summation of the following values:

- $\sum_{M' \neq \mathsf{Dead} \neq N'} \sum_{i \in I_1, j \in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(M', N') \cdot d(M', N')$
- $\sum_{M' \neq \mathsf{Dead}} \sum_{i \in I_1, j \in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(M', \mathsf{Dead}) \cdot d(M', \mathsf{Dead}) + \sum_{i \in I_1, j \in J_2} \omega(M_i, N_j) \cdot \gamma_{i,j}(M') \cdot d(M', \mathsf{Dead})$
- $\sum_{N' \neq \mathsf{Dead}} \sum_{i \in I_1, j \in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(\mathsf{Dead}, N') \cdot d(\mathsf{Dead}, N') + \sum_{i \in I_2, j \in J_1} \omega(M_i, N_j) \cdot \gamma'_{i,j}(N') \cdot d(\mathsf{Dead}, N')$
- $\sum_{i \in I_1, j \in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(\mathsf{Dead}, \mathsf{Dead}) \cdot d(\mathsf{Dead}, \mathsf{Dead}) + \sum_{i \in I_1, j \in J_2} \omega(M_i, N_j) \cdot \gamma_{i,j}(\mathsf{Dead}) \cdot d(\mathsf{Dead}, \mathsf{Dead})$ $+ \sum_{i \in I_2, j \in J_1} \omega(M_i, N_j) \cdot \gamma'_{i,j}(\mathsf{Dead}) \cdot d(\mathsf{Dead}, \mathsf{Dead}) + \sum_{i \in I_2, j \in J_2} \omega(M_i, N_j) \cdot d(\mathsf{Dead}, \mathsf{Dead})$.

By moving the first summand of the second, third and fourth items to the first item, we rewrite this summation as the summation of the following values:

- $\sum_{M', N' \in \mathtt{pCCPS}} \sum_{i \in I_1, j \in J_1} \omega(M_i, N_j) \cdot \omega_{i,j}(M', N') \cdot d(M', N')$
- $\sum_{i \in I_1, j \in J_2} \omega(M_i, N_j) \cdot \gamma_{i,j}(M') \cdot d(M', \mathsf{Dead})$
- $\sum_{i \in I_2, j \in J_1} \omega(M_i, N_j) \cdot \gamma'_{i,j}(N') \cdot d(\mathsf{Dead}, N')$
- $\sum_{i \in I_1, j \in J_2} \omega(M_i, N_j) \cdot \gamma_{i,j}(\mathsf{Dead}) \cdot d(\mathsf{Dead}, \mathsf{Dead}) + \sum_{i \in I_2, j \in J_1} \omega(M_i, N_j) \cdot \gamma'_{i,j}(N') \cdot d(\mathsf{Dead}, \mathsf{Dead}) + \sum_{i \in I_2, j \in J_2} \omega(M_i, N_j) \cdot d(\mathsf{Dead}, \mathsf{Dead})$.

By the definition of $\omega_{i,j}$ the first item is $\sum_{i \in I_1, j \in J_1} \omega(M_i, N_j) \cdot \mathbf{K}(d)(\gamma_{i,j}, \gamma'_{i,j})$. If $d(M_i, N_j) < 1$, we chosen $\gamma'_{i,j}$ such that $\mathbf{K}(d)(\gamma_{i,j}, \gamma'_{i,j}) \leq d(M_i, N_j)$. If $d(M_i, N_j) = 1$, then $\mathbf{K}(d)(\gamma_{i,j}, \gamma'_{i,j}) \leq d(M_i, N_j)$ is immediate. Henceforth we are sure that in all cases the first item is less or equal $\sum_{i \in I_1, j \in J_1} \omega(M_i, N_j) \cdot d(M_i, N_j)$. The second item is clearly less or equal than $\sum_{i \in I_1, j \in J_2} \omega(M_i, N_j)$. The third item is clearly less or equal than $\sum_{i \in I_2, j \in J_1} \omega(M_i, N_j)$. Finally, the last item is 0 since $d(\mathsf{Dead}, \mathsf{Dead}) = 0$. Summarising, we have $\sum_{M', N' \in \mathtt{pCCPS}} \omega'(M', N') \cdot d(M', N') \leq \sum_{i \in I_1, j \in J_1} \omega(M_i, N_j) \cdot d(M_i, N_j) + \sum_{i \in I_1, j \in J_2} \omega(M_i, N_j) + \sum_{i \in I_2, j \in J_1} \omega(M_i, N_j)$. Since $\mathbf{K}(d)(\rho_M, \rho_N)$ is the summation of the following values:

- $\sum_{i \in I_1, j \in J_1} \omega(M_i, N_j) \cdot d(M_i, N_j)$
- $\sum_{i \in I_1, j \in J_2} \omega(M_i, N_j) \cdot d(M_i, N_j) = \sum_{i \in I_1, j \in J_2} \omega(M_i, N_j)$ ($M_i \xrightarrow{\beta_2}$ and $N_j \not\stackrel{\hat{\beta}_2}{\Longrightarrow}$ give $d(M_i, N_j) = 1$)
- $\sum_{i \in I_2, j \in J_1} \omega(M_i, N_j) \cdot d(M_i, N_j) = \sum_{i \in I_2, j \in J_1} \omega(M_i, N_j)$ ($N_j \xrightarrow{\beta_2}$ and $M_i \not\stackrel{\hat{\beta}_2}{\Longrightarrow}$ give $d(M_i, N_j) = 1$)
- $\sum_{i \in I_2, j \in J_2} \omega(M_i, N_j) \cdot d(M_i, N_j)$.

it follows $\sum_{i \in I_1, j \in J_1} \omega(M_i, N_j) \cdot d(M_i, N_j) + \sum_{i \in I_1, j \in J_2} \omega(M_i, N_j) + \sum_{i \in I_2, j \in J_1} \omega(M_i, N_j) \leq \mathbf{K}(d)(\rho_M, \rho_N)$. Since we had $\mathbf{K}(d)(\rho_M, \rho_N) \leq d(M, N)$ we can conclude $\sum_{M', N' \in \mathtt{pCCPS}} \omega'(M', N') \cdot d(M', N') \leq d(M, N)$, as required. □

We are now ready to prove that all $\mathbf{d}^n$ are pseudometrics.

Proof of Proposition 5.8 We have to prove that $\mathbf{d}^n(M, M) = 0$, $\mathbf{d}^n(M, N) = \mathbf{d}^n(N, M)$ and $\mathbf{d}^n(M, N) \leq \mathbf{d}^n(M, O) + \mathbf{d}^n(O, N)$ for all $M, N, O \in \mathtt{pCCPS}$. We reason by induction over n. The base case $n = 0$ is immediate since $\mathbf{d}^0(M, N) = 0$ for all $M, N \in \mathtt{pCCPS}$. We consider the inductive step $n + 1$.

Let us start by proving $\mathbf{d}^{n+1}(M, M) = 0$. We have to show that for each transition $M \xrightarrow{\alpha} \gamma$ there is a transition $M \stackrel{\hat{\alpha}}{\Longrightarrow} \rho$ with $\mathbf{K}(\mathbf{d}^n)(\gamma, \rho + (1 - |\rho|)\overline{\mathsf{Dead}}) = 0$. We choose $\rho = \gamma$ and the transition $M \xrightarrow{\alpha} \gamma$. We obtain $\mathbf{K}(\mathbf{d}^n)(\gamma, \rho + (1 - |\rho|)\overline{\mathsf{Dead}}) = \mathbf{K}(\mathbf{d}^n)(\gamma, \gamma) = 0$, with the last equality by the inductive hypothesis and Lemma Appendix A.3.

The symmetry $\mathbf{d}^{n+1}(M, N) = \mathbf{d}^{n+1}(N, M)$ follows by $\mathbf{d}^{n+1}(M, N) = \mathbf{B}(\mathbf{d}^n)(M, N) = \mathbf{B}(\mathbf{d}^n)(N, M) = \mathbf{d}^{n+1}(N, M)$, where the second equality follows immediately by the definition of $\mathbf{B}$.

Finally we prove the triangular property $\mathbf{d}^{n+1}(M,N) \leq \mathbf{d}^{n+1}(M,O) + \mathbf{d}^{n+1}(O,N)$. This result is immediate if $\mathbf{d}^{n+1}(M,O) = 1$ or $\mathbf{d}^{n+1}(O,N) = 1$. Otherwise, it is enough to prove that any $M \xrightarrow{\alpha} \gamma_M$ is mimicked by some transition $N \stackrel{\hat{\alpha}}{\Longrightarrow} \gamma_N$ with $\mathbf{K}(\mathbf{d}^n)(\gamma_M, \gamma_N + (1- |\gamma_N|)\overline{\mathsf{Dead}}) \leq \mathbf{d}^{n+1}(M,O) + \mathbf{d}^{n+1}(O,N)$. From $M \xrightarrow{\alpha} \gamma_M$ and $\mathbf{d}^{n+1}(M,O) < 1$ we immediately infer that there is a transition $O \stackrel{\hat{\alpha}}{\Longrightarrow} \gamma_O$ with $\mathbf{K}(\mathbf{d}^n)(\gamma_M, \gamma_O + (1- |\gamma_O|)\overline{\mathsf{Dead}}) \leq \mathbf{d}^{n+1}(M,O)$. By Lemma Appendix A.4, from $O \stackrel{\hat{\alpha}}{\Longrightarrow} \gamma_O$ and $\mathbf{d}^{n+1}(O,N) < 1$ there is a transition $N \stackrel{\hat{\alpha}}{\Longrightarrow} \gamma_N$ such that $\mathbf{K}(\mathbf{d}^n)(\gamma_O + (1- |\gamma_O|)\overline{\mathsf{Dead}}, \gamma_N + (1- |\gamma_N|)\overline{\mathsf{Dead}}) \leq \mathbf{d}^{n+1}(O,N)$. By the inductive hypothesis and Lemma Appendix A.3 we get that $\mathbf{K}(\mathbf{d}^n)$ is a pseudometric, hence it satisfies the triangle inequality, namely $\mathbf{K}(\mathbf{d}^n)(\gamma_M, \gamma_N + (1- |\gamma_N|)\overline{\mathsf{Dead}}) \leq \mathbf{K}(\mathbf{d}^n)(\gamma_M, \gamma_O + (1- |\gamma_O|)\overline{\mathsf{Dead}}) + \mathbf{K}(\mathbf{d}^n)(\gamma_O + (1- |\gamma_O|)\overline{\mathsf{Dead}}, \gamma_N + (1- |\gamma_N|)\overline{\mathsf{Dead}})$. Therefore we can conclude the proof by $\mathbf{K}(\mathbf{d}^n)(\gamma_M, \gamma_N + (1- |\gamma_N|)\overline{\mathsf{Dead}}) \leq \mathbf{K}(\mathbf{d}^n)(\gamma_M, \gamma_O + (1- |\gamma_O|)\overline{\mathsf{Dead}}) + \mathbf{K}(\mathbf{d}^n)(\gamma_O + (1- |\gamma_O|)\overline{\mathsf{Dead}}, \gamma_N + (1- |\gamma_N|)\overline{\mathsf{Dead}}) \leq \mathbf{d}^{n+1}(M,O) + \mathbf{d}^{n+1}(O,N)$. □

In order to prove the compositionality or our weak bisimilarity metrics, i.e. Theorem 5.11, we divide its statement in six different propositions. To prove that $\approx_p$ preserves the compositionality we need a number of technical lemmas.

Given a distribution γ over CPSs and a CPS O, we denote with $\gamma \uplus O$ the distribution defined by $(\gamma \uplus O)(M \uplus O) = \gamma(M)$ for all CPSs M.

Lemma Appendix A.5 serves to propagate untimed actions on parallel CPSs.

Lemma Appendix A.5. *Assume two physically disjoint CPSs M_1 and M_2 such that $M_2 = E_2; S_2 \bowtie P_2$ and $E_2 = \langle evol^2, meas^2, inv^2 \rangle$. If $M_1 \xrightarrow{\alpha} \gamma$, with $\alpha \neq \mathsf{tick}$, and $S_2 \in inv^2$ then $M_1 \uplus M_2 \xrightarrow{\alpha} \gamma \uplus M_2$.*

Proof. If M_1 is the CPS Dead then also $M_1 \uplus M_2$ is Dead and the thesis is immediate. Consider the case $M_1 \neq \mathsf{Dead}$. Let us assume that $M_1 = E_1; S_1 \bowtie P_1$ with $E_1 = \langle evol^1, meas^1, inv^1 \rangle$ and $S_1 = \langle \xi^1_{\mathrm{x}}, \xi^1_{\mathrm{s}}, \xi^1_{\mathrm{a}} \rangle$. Moreover, assume that $S_2 = \langle \xi^2_{\mathrm{x}}, \xi^2_{\mathrm{s}}, \xi^2_{\mathrm{a}} \rangle$. We consider the case in which $M_1 \xrightarrow{\alpha} \gamma$ is derived by rule (SensRead). The other cases where the transition is derived by the other rules in Table 2 can be proved in a similar manner. In this case, we have $\alpha = \tau$ and there are a sensor s, probability values p_i and real values v_i with $i \in I$ and a distribution π such that the rule (SensRead) instances as

$$\frac{P_1 \xrightarrow{s?(z)} \pi \qquad \xi^1_{\mathrm{s}}(s) = \sum_{i \in I} p_i \cdot \overline{v_i} \qquad \xi^1_{\mathrm{x}} \in inv^1}{\langle \xi^1_{\mathrm{x}}, \xi^1_{\mathrm{s}}, \xi^1_{\mathrm{a}} \rangle \bowtie P_1 \xrightarrow{\tau} \overline{\langle \xi^1_{\mathrm{x}}, \xi^1_{\mathrm{s}}, \xi^1_{\mathrm{a}} \rangle} \bowtie \sum_{i \in I} p_i \cdot \pi\{^{v_i}/_z\}}$$

and $\gamma = E_1; \overline{\langle \xi^1_{\mathrm{x}}, \xi^1_{\mathrm{s}}, \xi^1_{\mathrm{a}} \rangle} \bowtie \sum_{i \in I} p_i \cdot \pi\{^{v_i}/_z\}$.

Now we argue that we can apply rule (SensRead) to infer a transition by $M_1 \uplus M_2$. Recall that $M_1 \uplus M_2$ is the CPS $(E_1 \uplus E_2); \langle \xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}}, \xi^1_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}}, \xi^1_{\mathrm{a}} \uplus \xi^2_{\mathrm{a}} \rangle \bowtie P_1 \parallel P_2$. Let $E_1 \uplus E_2 = \langle evol, meas, inv \rangle$. From $P_1 \xrightarrow{s?(z)} \pi$, by rule (Par) in Table 1 we can derive the transition $P_1 \parallel P_2 \xrightarrow{s?(z)} \pi \parallel \overline{P_2}$, which is one of the premises of rule (SensRead) necessary to infer a transition by $\langle \xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}}, \xi^1_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}}, \xi^1_{\mathrm{a}} \uplus \xi^2_{\mathrm{a}} \rangle \bowtie P_1 \parallel P_2$. Then, the premise $\xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}} \in inv$ of (SensRead) follows by $\xi^1_{\mathrm{x}} \in inv^1$, the hypothesis $\xi^2_{\mathrm{x}} \in inv^2$ and the property $\xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}} \in inv$ iff $\xi^1_{\mathrm{x}} \in inv^1$ and $\xi^2_{\mathrm{x}} \in inv^2$. Finally, the premise $(\xi^1_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}})(s) = \sum_{i \in I} p_i \cdot \overline{v_i}$ follows by $(\xi^1_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}})(s) = \xi^1_{\mathrm{s}}(s)$ and $\xi^1_{\mathrm{s}}(s) = \sum_{i \in I} p_i \cdot \overline{v_i}$. Therefore we have

$$\frac{P_1 \parallel P_2 \xrightarrow{s?(z)} \pi \parallel \overline{P_2} \qquad (\xi^1_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}})(s) = \sum_{i \in I} p_i \cdot \overline{v_i} \qquad \xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}} \in inv}{\langle \xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}}, \xi^1_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}}, \xi^1_{\mathrm{a}} \uplus \xi^2_{\mathrm{a}} \rangle \bowtie P_1 \parallel P_2 \xrightarrow{\tau} \overline{\langle \xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}}, \xi^1_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}}, \xi^1_{\mathrm{a}} \uplus \xi^2_{\mathrm{a}} \rangle} \bowtie \sum_{i \in I} p_i \cdot (\pi \parallel \overline{P_2})\{^{v_i}/_z\}}$$

with $(E_1 \uplus E_2); \overline{\langle \xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}}, \xi^1_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}}, \xi^1_{\mathrm{a}} \uplus \xi^2_{\mathrm{a}} \rangle} \bowtie \sum_{i \in I} p_i \cdot (\pi \parallel \overline{P_2})\{^{v_i}/_z\} = \gamma \uplus M_2$. □

Lemma Appendix A.5 can be generalised to weak transitions.

Lemma Appendix A.6. *Assume two physically disjoint CPSs M_1 and M_2 such that $M_2 = E_2; S_2 \bowtie P_2$ and $E_2 = \langle evol^2, meas^2, inv^2 \rangle$. If $M_1 \stackrel{\widehat{\alpha}}{\Longrightarrow} \gamma$, with $\alpha \neq \mathsf{tick}$, and $S_2 \in inv^2$ then $M_1 \uplus M_2 \stackrel{\widehat{\alpha}}{\Longrightarrow} \gamma \uplus M_2$.*

Proof. By induction over the length n of $\stackrel{\widehat{\alpha}}{\Longrightarrow}$. The base case $n = 1$ is given by Lemma Appendix A.5. Consider the inductive step $n + 1$. We have $M_1 \stackrel{\widehat{\alpha_1}}{\Longrightarrow} \gamma' \xrightarrow{\widehat{\alpha_2}} \gamma$ with either $\alpha_1 = \alpha$ and $\alpha_2 = \tau$, or $\alpha_1 = \tau$ and $\alpha_2 = \alpha$. Since the

length of $\stackrel{\widehat{\alpha_1}}{\Longrightarrow}$ is n, we can apply the inductive hypothesis and infer $M_1 \uplus M_2 \stackrel{\widehat{\alpha_1}}{\Longrightarrow} \gamma' \uplus M_2$. Assume $\gamma' = \sum_{i\in I} p_i \cdot \overline{M_i}$, for suitable probability values p_i and CPS M_i. By definition, $\gamma' \xrightarrow{\widehat{\alpha_2}} \gamma$ implies that there exists a subset $J \subseteq I$ with $M_j \xrightarrow{\widehat{\alpha_2}} \gamma_j$ for all $j \in J$, $M_i \stackrel{\alpha_2}{\not\longrightarrow}$ for $i \in I \setminus J$ and $\gamma = \sum_{j\in J} p_j \cdot \overline{M_j}$. We can prove now that for any $j \in J$ we have $M_j \uplus M_2 \xrightarrow{\widehat{\alpha_2}} \gamma_j \uplus M_2$. We distinguish two cases. The first case is $M_j \xrightarrow{\alpha_2} \gamma_j$. By Lemmma Appendix A.5 we get $M_j \uplus M_2 \xrightarrow{\alpha_2} \gamma_j \uplus M_2$, and, therefore, $M_j \uplus M_2 \xrightarrow{\widehat{\alpha_2}} \gamma_j \uplus M_2$. The second case is $\alpha_2 = \tau$ and $\gamma_j = \overline{M_j}$. We immediately have $M_j \uplus M_2 \xrightarrow{\widehat{\tau}} \gamma_j \uplus M_2$. Hence $\sum_{j\in J} M_j \uplus M_2 \xrightarrow{\widehat{\alpha_2}} \sum_{j\in J} \gamma_j \uplus M_2$, namely $\gamma' \uplus M_2 \xrightarrow{\widehat{\alpha_2}} \gamma \uplus M_2$. Then, from $M \uplus M_2 \stackrel{\widehat{\alpha_1}}{\Longrightarrow} \gamma' \uplus M_2$ and $\gamma' \uplus M_2 \xrightarrow{\widehat{\alpha_2}} \gamma \uplus M_2$ we get $M \uplus M_2 \stackrel{\widehat{\alpha}}{\Longrightarrow} \gamma \uplus M_2$, which completes the proof. □

Next lemma says that the invariants of CPSs in distance < 1 must agree.

Lemma Appendix A.7. *Assume two CPSs M_1 and M_2 such that $M_i = E_i; S_i \bowtie P_i$ and $E_i = \langle evol^i, meas^i, inv^i\rangle$, for $i = 1, 2$. If $\mathbf{d}(M_1, M_2) < 1$ then $S_1 \in inv^1$ iff $S_2 \in inv^2$.*

Proof. The proof is by contradiction. Assume that $\mathbf{d}(M_1, M_2) < 1$, $S_1 \in inv^1$ and $S_2 \notin inv^2$. We show that $M_1 \stackrel{\widehat{\mathsf{tick}}}{\Longrightarrow}$ and $M_2 \stackrel{\widehat{\mathsf{tick}}}{\Longrightarrow\!\!\!\!\!\!/}$, which contradicts $\mathbf{d}(M_1, M_2) < 1$. By the well timedness property for CPSs (Theorem 2.9, last item), there exists a natural n such that all derivations $M_1 \xrightarrow{\tau} N_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} N_k$ are such that $k \le n$, then we have $N_k \stackrel{\tau}{\not\longrightarrow}$. Since $N_k \stackrel{\tau}{\not\longrightarrow}$, by the maximal progress property for CPSs (Theorem 2.9, second item) it follows that $N_k \xrightarrow{\mathsf{tick}} \gamma$, for some γ. We conclude $M_1 \stackrel{\widehat{\mathsf{tick}}}{\Longrightarrow}$. Since $S_2 \notin inv^2$, the CPS M_2 can perform only the step $M_2 \xrightarrow{\tau} \mathsf{Dead}$ and Dead can not perform any action, and hence, $M_2 \stackrel{\widehat{\mathsf{tick}}}{\Longrightarrow\!\!\!\!\!\!/}$. □

Here comes one of the main technical result: the bisimilarity metric is preserved by the parallel composition of physically disjoint CPSs.

Proposition Appendix A.8. $\mathbf{d}(M \uplus O, N \uplus O) \le \mathbf{d}(M, N)$*, for any physically disjoint CPS O.*

Proof. The case $\mathbf{d}(M, N) = 1$ is immediate, therefore we assume $\mathbf{d}(M, N) < 1$. Let us define the function $d\colon \mathtt{pCCPS} \times \mathtt{pCCPS} \to [0, 1]$ by $d(M \uplus O, N \uplus O) = \mathbf{d}(M, N)$ for all $M, N, O \in \mathtt{pCCPS}$. To prove the thesis it is enough to show that d is a weak bisimulation metric. In fact, since $\mathbf{d}$ is the minimal weak bisimulation metric, we infer $\mathbf{d} \sqsubseteq d$, thus giving $\mathbf{d}(M \uplus O, N \uplus O) \le d(M \uplus O, N \uplus O) = \mathbf{d}(M, N)$. To prove that d is a weak bisimulation metric, we show that any transition $M \uplus O \xrightarrow{\alpha} \gamma$ is simulated by some transition $N \uplus O \stackrel{\widehat{\alpha}}{\Longrightarrow} \gamma'$ with $\mathbf{K}(d)(\gamma, \gamma' + (1 - |\gamma'|)\mathsf{Dead}) \le d(M \uplus O, N \uplus O)$. The cases where one of the CPSs M, N and O are Dead is immediate. Hence, assume that M, N and O are not Dead. Let us assume that $M_1 = E_1; S_1 \bowtie P_1$ with $E_1 = \langle evol^1, meas^1, inv^1\rangle$ and $S_1 = \langle \xi^1_{\mathrm{x}}, \xi^1_{\mathrm{s}}, \xi^1_{\mathrm{a}}\rangle$. Moreover, assume that $O = E_2; S_2 \bowtie P_2$ with $E_2 = \langle evol^2, meas^2, inv^2\rangle$ and $S_2 = \langle \xi^2_{\mathrm{x}}, \xi^2_{\mathrm{s}}, \xi^2_{\mathrm{a}}\rangle$. Finally $E_1 \uplus E_2 = \langle evol, meas, inv\rangle$.

We proceed by case analysis on how $M \uplus O \xrightarrow{\alpha} \gamma$ is derived. The cases are the following:

- The transition $M \uplus O \xrightarrow{\tau} \gamma$ is derived by rule (SensRead) in Table 2, instantiated as

$$\frac{P_1 \parallel P_2 \xrightarrow{s?(z)} \pi \qquad (\xi^1_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}})(s) = \sum_{i\in I} p_i \cdot \overline{\nu_i} \qquad \xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}} \in inv}{S_1 \uplus S_2 \bowtie P_1 \parallel P_2 \xrightarrow{\tau} \overline{S_1 \uplus S_2} \bowtie \sum_{i\in I} p_i \cdot \pi\{^{\nu_i}/_z\}}$$

  with $\gamma = (E_1 \uplus E_2); \overline{S_1 \uplus S_2} \bowtie \sum_{i\in I} p_i \cdot \pi\{^{\nu_i}/_z\}$.

- The transition $M \uplus O \xrightarrow{\tau} \gamma$ is derived by rule (ActWrite) in Table 2 instantiated as

$$\frac{P_1 \parallel P_2 \xrightarrow{a!v} \pi \qquad \xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}} \in inv}{\langle \xi^1_{\mathrm{x}} \uplus \xi^2_{\mathrm{x}}, \xi^1_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}}, \xi^1_{\mathrm{a}} \uplus \xi^2_{\mathrm{a}}\rangle \bowtie P_1 \parallel P_2 \xrightarrow{\tau} \overline{\langle \xi^1_{\mathrm{x}} \uplus \xi^1_{\mathrm{x}}, \xi^1_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}}, \xi^2_{\mathrm{a}} \uplus \xi^1_{\mathrm{a}}[a \mapsto v]\rangle} \bowtie \pi}$$

- The transition $M \uplus O \xrightarrow{\tau} \gamma$ is derived by rule (Tau) in Table 2, instantiated as

$$\frac{P_1 \parallel P_2 \xrightarrow{\tau} \pi \qquad (S_1 \uplus S_2) \in inv}{S_1 \uplus S_2 \bowtie P_1 \parallel P_2 \xrightarrow{\tau} \overline{S_1 \uplus S_2} \bowtie \pi}$$

  with $\gamma = (E_1 \uplus E_2); \overline{S_1 \uplus S_2} \bowtie \pi$.

- The transition $M \uplus O \xrightarrow{\mathsf{tick}} \gamma$ is derived by rule (Time) in Table 2, instantiated as

$$\frac{P_1 \parallel P_2 \xrightarrow{\mathsf{tick}} \pi \qquad S_1 \uplus S_2 \bowtie P_1 \parallel P_2 \xrightarrow{\tau} \!\!\!\!/ \qquad (S_1 \uplus S_2) \in inv}{S_1 \uplus S_2 \bowtie P_1 \parallel P_2 \xrightarrow{\mathsf{tick}} next_{(E_1 \uplus E_2)}(S_1 \uplus S_2) \bowtie \pi}$$

  with $\gamma = (E_1 \uplus E_2) : next_{(E_1 \uplus E_2)}(S_1 \uplus S_2) \bowtie \pi$.

- The transition $M \uplus O \xrightarrow{cv} \gamma$ is derived by rule (Inp) in Table 2, instantiated as

$$\frac{P_1 \parallel P_2 \xrightarrow{cv} \pi \quad (S_1 \uplus S_2) \in inv}{S_1 \uplus S_2 \bowtie P_1 \parallel P_2 \xrightarrow{cv} \overline{S_1 \uplus S_2} \bowtie \pi}$$

  with $\gamma = (E_1 \uplus E_2); \overline{S_1 \uplus S_2} \bowtie \pi$.

- The transition $M \uplus O \xrightarrow{\bar{cv}} \gamma$ is derived by rule (Out) in Table 2 instantiated as

$$\frac{P_1 \parallel P_2 \xrightarrow{\bar{cv}} \pi \qquad S_1 \uplus S_2 \in inv}{S_1 \uplus S_2 \bowtie P_1 \parallel P_2 \xrightarrow{\bar{cv}} \overline{S_1 \uplus S_2} \bowtie \pi}.$$

We show only the first case, the other are analogous. We recall that, by definition of operator $\uplus$, the physical environments E_1 and E_2 have different physical devices. Thus, there are two cases:

- s is a sensor of E_1. In this case, the transition $P_1 \parallel P_2 \xrightarrow{s?(z)} \pi$ derives by rule (Par) in Table 1 from $P_1 \xrightarrow{s?(z)} \pi'$, where π' is a process distribution such that $\pi = \pi' \parallel \overline{P_2}$.

  First we argue that rule (SensRead) can be used to derive a transition by M. From $(S_1 \uplus S_2) \in inv$, by definition of $E_1 \uplus E_2$, we get both $S_1 \in inv^1$ and $S_2 \in inv^2$. From $(\xi_{\mathrm{s}}^1 \uplus \xi_{\mathrm{s}}^2)(s) = \sum_{i \in I} p_i \cdot \overline{v_i}$, since s is a sensor of ξ_{s}^1, we derive $\xi_{\mathrm{s}}^1(s) = \sum_{i \in I} p_i \cdot \overline{v_i}$. Summarising, we have $P_1 \xrightarrow{s?(z)} \pi'$, $S_1 \in inv^1$, and $\xi_{\mathrm{s}}^1(s) = \sum_{i \in I} p_i \cdot \overline{v_i}$, which allows us to apply rule (SensRead) and derive $S_1 \bowtie P_1 \xrightarrow{\tau} \overline{S_1} \bowtie \sum_{i \in I} p_i \cdot (\pi')\{{}^{v_i}/_{z}\}$, namely $M \xrightarrow{\tau} \gamma'' = E_1; \overline{S_1} \bowtie \sum_{i \in I} p_i \cdot (\pi')\{{}^{v_i}/_{z}\}$.

  Then, from transition $M \xrightarrow{\tau} \gamma''$ and $\mathbf{d}(M, N) < 1$, there is a distribution γ''' such that $N \stackrel{\widehat{\tau}}{\Longrightarrow} \gamma'''$ with $\mathbf{K}(\mathbf{d})(\gamma'', \gamma''' + (1 - |\gamma'''|)\mathsf{Dead}) \leq \mathbf{d}(M, N)$. Since $S_2 \in inv^2$, by Lemma Appendix A.6 it follows that $N \uplus O \stackrel{\widehat{\tau}}{\Longrightarrow} \gamma''' \uplus O$. Finally, we conclude that $\gamma''' \uplus O$ is the distribution γ' we were looking for by $\mathbf{K}(d)(\gamma, \gamma''' \uplus O + (1 - |\gamma''' \uplus O|)\mathsf{Dead}) = \mathbf{K}(d)(\gamma'' \uplus O, \gamma''' \uplus O + (1 - |\gamma''' \uplus O|)\mathsf{Dead}) = \mathbf{K}(\mathbf{d})(\gamma'', \gamma'''(1 - |\gamma'''|)\mathsf{Dead}) \leq \mathbf{d}(M, N) = d(M \uplus O, N \uplus O)$.

- s is a sensor of E_2. In this case, the transition $P_1 \parallel P_2 \xrightarrow{s?(z)} \pi$ derives by rule (Par) in Table 1 from $P_2 \xrightarrow{s?(z)} \pi'$, where π' is a process distribution such that $\pi = \overline{P_1} \parallel \pi'$.

  Assume $N = E_3; S_3 \bowtie P_3$ with $E_3 = \langle evol^3, meas^3, inv^3 \rangle$ and $S_3 = \langle \xi_{\mathrm{x}}^3, \xi_{\mathrm{s}}^3, \xi_{\mathrm{a}}^3 \rangle$. We show that rule (SensRead) allow us to infer $N \uplus O \xrightarrow{\tau} N \uplus \gamma''$ for some γ''.

  By the rule (Par) we get $P_3 \parallel P_2 \xrightarrow{s?(z)} \overline{P_3} \parallel \pi'$. From $(S_1 \uplus S_2) \in inv$, by definition of $E_1 \uplus E_2$. we get both $S_1 \in inv^1$ and $S_2 \in inv^2$. Let $E_1 \uplus E_3 = \langle evol', meas', inv' \rangle$. From $\mathbf{d}(M, N) < 1$ and $S_1 \in inv^1$, by Lemma Appendix A.7 it follows that $S_3 \in inv^3$ and so $(S_3 \uplus S_2) \in inv'$. From $(\xi_{\mathrm{s}}^1 \uplus \xi_{\mathrm{s}}^2)(s) = \sum_{i \in I} p_i \cdot \overline{v_i}$, since

s is a sensor of ξ^2_{s}, we derive $\xi^2_{\mathrm{s}}(s) = \sum_{i\in I} p_i \cdot \overline{v_i}$. Hence we derive $(\xi^3_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}})(s) = \sum_{i\in I} p_i \cdot \overline{v_i}$. Summarising we have $P_3 \parallel P_2 \xrightarrow{s?(z)} \overline{P_3} \parallel \pi'$, $(S_3 \uplus S_2) \in inv'$ and $(\xi^3_{\mathrm{s}} \uplus \xi^2_{\mathrm{s}})(s) = \sum_{i\in I} p_i \cdot \overline{v_i}$. Hence, we can apply rule (SensRead) to infer $N \uplus O \xrightarrow{\tau} (E_3 \uplus E_2); \overline{S_3 \uplus S_2} \bowtie \sum_{i\in I} p_i \cdot (\overline{P_3} \parallel \pi')\{{}^{v_i}/_z\} = N \uplus \gamma''$ with $\gamma'' = E_2; \overline{S_2} \bowtie \sum_{i\in I} p_i \cdot (\pi')\{{}^{v_i}/_z\}$. Finally, we can conclude that $\gamma' = N \uplus \gamma''$ is the distribution we were looking for by $\mathbf{K}(d)(M \uplus \gamma'', N \uplus \gamma'') = \mathbf{K}(\mathbf{d})(\overline{M}, \overline{N}) = \mathbf{d}(M, N) = d(M \uplus O, N \uplus O)$.

□

Also the n-weak bisimilarity metric is preserved by the parallel composition of physically disjoint CPSs.

Proposition Appendix A.9. *$\mathbf{d}^n(M \uplus O, N \uplus O) \leq \mathbf{d}^n(M, N)$, for any physically disjoint CPS O and $n \geq 0$.*

Proof. We proceed by induction over n. The base case $n = 0$ is immediate since $\mathbf{d}^n(M, N) = \mathbf{0}(M, N) = 0$ for all $M, N \in \mathtt{pCCPS}$. We consider the inductive step $n + 1$. The case $\mathbf{d}^{n+1}(M, N) = 1$ is immediate, therefore we assume $\mathbf{d}^{n+1}(M, N) < 1$. We have to show that any transition $M \uplus O \xrightarrow{\alpha} \gamma$ is simulated by some transition $N \uplus O \stackrel{\widehat{\alpha}}{\Longrightarrow} \gamma'$ with $\mathbf{K}(\mathbf{d}^n)(\gamma, \gamma' + (1 - |\gamma'|)\mathsf{Dead}) \leq \mathbf{d}^{n+1}(M \uplus O, N \uplus O)$. This can be shown precisely as in the proof of Proposition Appendix A.8. Essentially, we have to replace all occurrences of $\mathbf{d}(M, N)$ by $\mathbf{d}^{n+1}(M, N)$ and all occurrences of $\mathbf{K}(d)(\gamma, \gamma')$ and $\mathbf{K}(\mathbf{d})(\gamma, \gamma')$ by $\mathbf{K}(\mathbf{d}^n)(\gamma, \gamma')$. □

Now we prove that our weak bisimilarity metrics are preserved by parallel composition of pure-logical processes. These are special cases of Proposition Appendix A.8 and Proposition Appendix A.9.

Proposition Appendix A.10. *$\mathbf{d}(M \parallel P, N \parallel P) \leq \mathbf{d}(M, N)$, for any pure-logical process P.*

Proof. Let $E_\emptyset$ be the physical environment with an empty set of state variables, sensors and actuators. Let $S_\emptyset$ be the unique (empty) physical state of $E_\emptyset$. We have $\mathbf{d}(M \parallel P, N \parallel P) \leq \mathbf{d}(M \parallel P, M \uplus (E_\emptyset; S_\emptyset \bowtie P)) + \mathbf{d}(M \uplus (E_\emptyset; S_\emptyset \bowtie P), N \parallel P) = \mathbf{d}(M \uplus (E_\emptyset; S_\emptyset \bowtie P), N \parallel P) \leq \mathbf{d}(M \uplus (E_\emptyset; S_\emptyset \bowtie P), N \uplus (E_\emptyset; S_\emptyset \bowtie P)) + \mathbf{d}(N \uplus (E_\emptyset; S_\emptyset \bowtie P), N \parallel P) = \mathbf{d}(M \uplus (E_\emptyset; S_\emptyset \bowtie P), N \uplus (E_\emptyset; S_\emptyset \bowtie P)) \leq \mathbf{d}(M, N)$ where the first two inequalities follow by the triangular properties of $\mathbf{d}$, the last inequality follows by Proposition Appendix A.8 and the two equalities are immediate. □

Proposition Appendix A.11. *$\mathbf{d}^n(M \parallel P, N \parallel P) \leq \mathbf{d}^n(M, N)$, for any pure-logical process P and $n \geq 0$.*

Proof. The same arguments used in the proof of Proposition Appendix A.10 apply. Essentially, we simply exploits Proposition Appendix A.9 instead of Proposition Appendix A.8. □

Finally, we prove that weak bisimilarity metrics are preserved by channel restriction.

Proposition Appendix A.12. *$\mathbf{d}(M\backslash c, N\backslash c) \leq \mathbf{d}(M, N)$, for any channel c.*

Proof. We reason as in Proposition Appendix A.8. The case $\mathbf{d}(M, N) = 1$ is immediate, therefore we assume $\mathbf{d}(M, N) < 1$. Let us define the function $d\colon \mathtt{pCCPS}\times\mathtt{pCCPS} \to [0, 1]$ by $d(M\backslash c, N\backslash c) = \mathbf{d}(M, N)$ for all $M, N, O \in \mathtt{pCCPS}$. To prove the thesis it is enough to show that d is a weak bisimulation metric. In fact, since $\mathbf{d}$ is the minimal weak bisimulation metric, this implies $\mathbf{d} \sqsubseteq d$, thus giving $\mathbf{d}(M\backslash c, N\backslash c) \leq d(M\backslash c, N\backslash c) = \mathbf{d}(M, N)$. To prove that d is a weak bisimulation metric, we show that any transition $M\backslash c \xrightarrow{\alpha} \gamma$ is simulated by some transition $N\backslash c \stackrel{\widehat{\alpha}}{\Longrightarrow} \gamma'$ with $\mathbf{K}(d)(\gamma, \gamma' + (1 - |\gamma'|)\mathsf{Dead}) \leq d(M \setminus c, N \setminus c)$. The proof proceeds by case analysis on why $M \setminus c \xrightarrow{\alpha} \gamma$. □

Proposition Appendix A.13. *$\mathbf{d}^n(M\backslash c, N\backslash c) \leq \mathbf{d}^n(M, N)$, for any channel c and $n \geq 0$.*

Proof. We reason as in Proposition Appendix A.9. Hence, we proceed by induction over n, where the base case $n = 0$ is immediate and we consider the inductive step $n + 1$. The case $\mathbf{d}^{n+1}(M, N) = 1$ is immediate, therefore we assume $\mathbf{d}^{n+1}(M, N) < 1$. We have to show that any transition $M\backslash c \xrightarrow{\alpha} \gamma$ is simulated by some transition $N\backslash c \stackrel{\widehat{\alpha}}{\Longrightarrow} \gamma'$ with $\mathbf{K}(\mathbf{d}^n)(\gamma, \gamma' + (1 - |\gamma'|)\mathsf{Dead}) \leq \mathbf{d}^{n+1}(M \setminus c, N \setminus c)$. The proof proceeds by case analysis on why $M \setminus c \xrightarrow{\alpha} \gamma$. □

Proof of Theorem 5.11 By Proposition Appendix A.8–Proposition Appendix A.13. □

Finally, as the bisimilarity $\approx$ coincides with the bisimulation metric $\approx_0$ it follows that Theorem 3.5 is a special case of Theorem 5.11.

Proof of Theorem 3.5 Consider the first result of Theorem 3.5. As $M \approx N$, by an application of Proposition 5.4 it follows that $\mathbf{d}(M, N) = 0$. By Theorem 5.11.1 we derive $\mathbf{d}(M \uplus O, N \uplus O) = 0$, for any physically-disjoint CPS O. By Proposition 5.4, it follows that $M \uplus O \approx N \uplus O$.

The proofs of the remaining two cases of Theorem 3.5 are analogous. □

Appendix A.4. Proofs of 6

Proof of Proposition 6.1 The proof is analogous to that of Proposition 6.2 and Proposition 6.3.1. □

As the bisimilarity $\approx$ coincides with the bisimulation metric $\approx_0$ it follows that Proposition 4.3 is a special case of Proposition 6.1.

Proof of Proposition 4.3 Directly by Proposition 6.1 (first item) and Proposition 5.4. □

Proof of Proposition 6.2 Define the CPS NIL as $NIL = E_\emptyset; S_\emptyset \bowtie \mathsf{nil}$, where $E_\emptyset$ is the empty physical environment and $S_\emptyset$ the unique (empty) physical state of $E_\emptyset$. The only transition by NIL is $NIL \xrightarrow{\mathsf{tick}} \overline{NIL}$. By Proposition 4.1 and Theorem 2.9 (fourth item) we infer that $\mathbf{d}^n(Eng_g, NIL) = 0$. Therefore, by the triangular property of $\mathbf{d}^n$, to show the thesis $\mathbf{d}^n(Eng_g, \widehat{Eng_g}) \leq 1 - \left(1 - q_g(p_g)^5\right)^n$ we can show $\mathbf{d}^n(NIL, \widehat{Eng_g}) \leq 1 - \left(1 - q_g(p_g)^5\right)^n$.

The proof obligation $\mathbf{d}^n(NIL, \widehat{Eng_g}) \leq 1 - \left(1 - (p_g)^5\right)^n$ follows from the following nine properties, by observing that the system $\widehat{Eng_g}$ satisfies the first one. In the following we denote the process $\mathsf{rec}\, Y.\mathsf{tick}^5.\mathsf{read}\, s_t(x).[x > 10]\{\mathsf{snd}\, warning\langle \mathrm{ID}\rangle.Y\}, \{\mathsf{write}\, cool\langle \mathsf{off}\rangle.\mathsf{tick}.Ctrl\}$ with $RecY$.

1. $\mathbf{d}^n(NIL, Env_g; S \bowtie P) \leq 1 - \left(1 - q_g(p_g)^5\right)^n$ whenever the physical state S satisfies $cool = \mathsf{off}$ and $temp \in [0, 10.1]$, and the process P is $Ctrl$, or $\mathsf{tick}.Ctrl$.

2. $\mathbf{d}^n(NIL, Env_g; S \bowtie P) \leq 1 - \left(1 - q_g(p_g)^5\right)^n$ whenever the physical state S satisfies $cool = \mathsf{off}$ and $temp \in (10.1, 11.4]$, and the process P is $Ctrl$, or $Cooling$.

3. $\mathbf{d}^n(NIL, Env_g; S \bowtie P) \leq 1 - \left(1 - (p_g)^5\right)\left(1 - q_g(p_g)^5\right)^n$ whenever the physical state S satisfies $cool = \mathsf{off}$ and $temp \in (10.4, 11.5]$, and the process P is $Ctrl$, or $Cooling$.

4. $\mathbf{d}^n(NIL, Env_g; S \bowtie P) \leq 1 - \left(1 - q_g(p_g)^5\right)^n$ whenever the physical state S satisfies $cool = \mathsf{on}$ and $temp \in (9.9, 11.4]$, and the process P is $RecY$.

5. $\mathbf{d}^n(NIL, Env_g; S \bowtie P) \leq 1 - \left(1 - (p_g)^5\right)\left(1 - q_g(p_g)^5\right)^n$ whenever the physical state S satisfies $cool = \mathsf{on}$ and $temp \in (10.4, 11.5]$, and the process P is $RecY$.

6. $\mathbf{d}^n(NIL, Env_g; S \bowtie P) \leq 1 - \left(1 - (p_g)^{5-k}\right)\left(1 - q_g(p_g)^5\right)^n$, for all $n \in [1, 4]$, whenever the physical state S satisfies $cool = \mathsf{on}$ and $temp \in (11.4 - k(0.3), 11.5 - k(0.3)]$, and the process P is

$$P = \mathsf{tick}^{5-k}.\mathsf{read}\, s_t(x)[x > 10]\{\mathsf{snd}\, warning\langle \mathrm{ID}\rangle.RecY\}, \{\mathsf{write}\, cool\langle \mathsf{off}\rangle.\mathsf{tick}.Ctrl\}.$$

7. $\mathbf{d}^n(NIL, Env_g; S \bowtie P) \leq 1 - \left(1 - q_g(p_g)^5\right)^n$ whenever the physical state S satisfies $cool = \mathsf{on}$ and $temp \leq 11.4 - k(0.3)$, and the process P is

$$P = \mathsf{tick}^{5-k}.\mathsf{read}\, s_t(x)[x > 10]\{\mathsf{snd}\, warning\langle \mathrm{ID}\rangle.RecY\}, \{\mathsf{write}\, cool\langle \mathsf{off}\rangle.\mathsf{tick}.Ctrl\}$$

for any $k \in [1, 4]$.

8. $\mathbf{d}^n(NIL, Env_g; S \bowtie P) \leq 1 - \left(1 - q_g(p_g)^5\right)^n$ whenever the physical state S satisfies $cool = \mathsf{on}$ and $temp \leq 9.9$, and the process P is

$$P = \mathsf{read}\, s_t(x)[x > 10]\{\mathsf{snd}\, warning\langle \mathrm{ID}\rangle.RecY\}, \{\mathsf{write}\, cool\langle \mathsf{off}\rangle.\mathsf{tick}.Ctrl\}.$$

9. $\mathbf{d}^n(NIL, Env_g; S \bowtie P) \leq 1 - \left(1 - q_g(p_g)^5\right)^n$ whenever the physical state S satisfies $cool = \mathsf{on}$ and $temp \leq 9.9$, and the process P is $P = \mathsf{write}\, cool\langle \mathsf{off}\rangle.\mathsf{tick}.Ctrl$.

We prove these nine properties in parallel, by induction over n. The base case $n = 0$ is immediate since $\mathbf{d}^0$ is the constant zero function $\mathbf{0}$. We consider the inductive step $n > 0$. First we observe that, given any distribution $\sum_{i\in I} p_i \cdot \overline{M_i}$ over CPS s, the only matching $\omega \in \Omega(\sum_{i\in I} p_i \cdot \overline{M_i}, \overline{NIL})$ is $\omega(M_i, NIL) = p_i$. It follows that $\mathbf{K}(\mathbf{d}^{n-1})(\sum_{i\in I} p_i \cdot \overline{M_i}, \overline{NIL}) = \sum_{i\in I} p_i \mathbf{d}^{n-1}(M_i, NIL)$. We show only the first property, the other are analogous.

We distinguish the cases $P = Ctrl$ and $P = \mathsf{tick}.Ctrl$.

- Case $P = Ctrl$.
  The only transition by $Env_g; S \bowtie P$ is $Env_g; S \bowtie P \xrightarrow{\tau} \sum_{i\in I} p_i \cdot \overline{M_i}$, where $M_i = Env_g; S \bowtie P_i$, with either $P_i = \mathsf{tick}.Ctrl$ or $P_i = Cooling$. The only transition by NIL is $NIL \xrightarrow{\tau} \overline{NIL}$. Therefore we infer $\mathbf{d}^n(Env_g; S \bowtie P, NIL) \leq \mathbf{K}(\mathbf{d}^{n-1})(\sum_{i\in I} p_i \cdot \overline{M_i}, \overline{NIL})$. By the inductive hypothesis on Proposition 1 we infer $\mathbf{d}^{n-1}(M_i, NIL) \leq 1 - \left(1 - q_g(p_g)^5\right)^{n-1}$ in both cases, thus implying

$$\mathbf{K}(\mathbf{d}^{n-1})(\sum_{i\in I} p_i \cdot \overline{M_i}, \overline{NIL}) = \sum_{i\in I} p_i \mathbf{d}^{n-1}(M_i, NIL) \leq 1 - \left(1 - q_g(p_g)^5\right)^{n-1} \leq 1 - \left(1 - q_g(p_g)^5\right)^n.$$

  This concludes the case.

- Case $P = \mathsf{tick}.Ctrl$.
  The only transition by $Env_g; S \bowtie P$ is $Env_g; S \bowtie P \xrightarrow{\mathsf{tick}} next_{Env_g;}(S) \bowtie \overline{Ctrl}$. Again, the only transition by NIL is $NIL \xrightarrow{\mathsf{tick}} \overline{NIL}$. Therefore $\mathbf{d}^n(Env_g; S \bowtie P, NIL) \leq \mathbf{K}(\mathbf{d}^{n-1})(next_{Env_g}(S) \bowtie \overline{Ctrl}, \overline{NIL})$. By definition, $next_{Env_g}(S) = \sum_{v\in[0.3,1.1]_g} \frac{1}{|[0.3,1.1]_g|} \overline{S[temp \mapsto \xi_{\mathrm{x}}(temp) - v]}$. Hence in all physical states S' in the support of $next_{Env_g}(S)$ we have $cool = \mathsf{off}$ and the temperature $temp$ lies in the interval $[0 + 0.3, 10.1 + 1.4]$.
  We have two cases: $temp \in [0+0.3, 10.1]$, and $temp \in (10.1, 10.5]$. If $temp \in [0+0.3, 10.1]$, then by the inductive hypothesis, case 1, we infer $\mathbf{d}^{n-1}(Env_g; S' \bowtie \overline{Ctrl}, NIL) \leq 1 - \left(1 - q_g(p_g)^5\right)^{n-1}$, for all $S' \in \mathsf{supp}(next_{Env_g}(S))$, thus implying

$$\mathbf{K}(\mathbf{d}^{n-1})(Env_g; next_E(S) \bowtie \overline{Ctrl}, \overline{NIL}) \leq 1 - \left(1 - q_g(p_g)^5\right)^{n-1} \leq 1 - \left(1 - q_g(p_g)^5\right)^n.$$

  If $temp \in (10.1, 10.5]$, then $temp \in (10.4, 10.5]$ with a probability bounded by q_g, whereas $temp \in (10.1, 10.4]$ with a probability not less that $1 - q_g$. If $temp \in (10.4, 10.5]$ we can apply the inductive hypothesis on Proposition 3 to get $\mathbf{d}^{n-1}(Env_g; S' \bowtie Ctrl, NIL) \leq 1 - \left(1 - (p_g)^5\right)\left(1 - q_g(p_g)^5\right)^{n-1}$, for all $S' \in \mathsf{supp}(next_{Env_g}(S))$. If $temp \in (10.4, 10.5]$ we can apply the inductive hypothesis on Proposition 2 to get $\mathbf{d}^{n-1}(Env_g; S' \bowtie Ctrl, NIL) \leq 1 - \left(1 - q_g(p_g)^5\right)^{n-1}$, for all $S' \in \mathsf{supp}(next_{Env_g}(S))$. Therefore for some $q \leq q_g$ we have

$$\begin{aligned}
& \mathbf{K}(\mathbf{d}^{n-1})(Env_g; next_E(S) \bowtie \overline{Ctrl}, \overline{NIL}) \\
= \;& (1-q)\left(1 - \left(1 - q_g(p_g)^5\right)^{n-1}\right) + q\left(1 - (p_g)^5\right)\left(1 - q_g(p_g)^5\right)^{n-1} \\
= \;& \left(1 - \left(1 - q_g(p_g)^5\right)^{n-1}\right) - q\left(1 - \left(1 - q_g(p_g)^5\right)^{n-1}\right) + q\left(1 - (p_g)^5\right)\left(1 - q_g(p_g)^5\right)^{n-1} \\
= \;& 1 - \left(1 - q_g(p_g)^5\right)^{n-1} - q + q\left(1 - q_g(p_g)^5\right)^{n-1} + q - \left(q - q(p_g)^5\right)\left(1 - q_g(p_g)^5\right)^{n-1} \\
= \;& 1 - q + q - \left(1 - q + q - q(p_g)^5\right)\left(1 - q_g(p_g)^5\right)^{n-1} \\
= \;& 1 - \left(1 - q(p_g)^5\right)\left(1 - q_g(p_g)^5\right)^{n-1} \\
\leq \;& 1 - \left(1 - q_g(p_g)^5\right)\left(1 - q_g(p_g)^5\right)^{n-1} \\
= \;& 1 - \left(1 - q_g(p_g)^5\right)^n
\end{aligned}$$

which completes the proof.

□

Proof of Proposition 6.3 By Proposition 6.2 we derive $\mathbf{d}^n(Eng_g, \widehat{Eng_g}) \leq 1 - \left(1 - q_g(p_g)^5\right)^n = p$. By simple α-conversion it follows that $\mathbf{d}^n(Eng_g^{\mathrm{L}}, \widehat{Eng_g^{\mathrm{L}}}) = p$ and $\mathbf{d}^n(Eng_g^{\mathrm{R}}, \widehat{Eng_g^{\mathrm{R}}}) = p$, respectively. By Theorem 5.11.4 (and the triangular property of $\mathbf{d}^n$) it follows that $\mathbf{d}^n(Eng_g^{\mathrm{L}} \uplus Eng_g^{\mathrm{R}}, \widehat{Eng_g^{\mathrm{L}}} \uplus \widehat{Eng_g^{\mathrm{R}}}) \leq 2p$. By Theorem 5.11.5 it follows that

$$\mathbf{d}^n\left(\left(Eng_g^{\mathrm{L}} \uplus (Eng_g^{\mathrm{R}}\right) \parallel Check, \left(\widehat{Eng_g^{\mathrm{L}}} \uplus (\widehat{Eng_g^{\mathrm{R}}}\right) \parallel Check\right) \leq 2p.$$

By Theorem 5.11.6 we obtain

$$\mathbf{d}^n\left(Airplane_g, \widehat{Airplane}_g\right) \leq 2p \tag{A.1}$$

thus confirming that Proposition 6.3.1 holds.

Finally, by Equation A.1 and Equation 1, we derive

$$\lim_{g \to +\infty} \mathbf{d}^n(Airplane_g, \widehat{Airplane}_g) \leq 2\left(1 - \left(1 - \frac{1}{8^6}\right)^n\right).$$

This proves Proposition 6.3.2. □

References

[1] R. Lanotte, M. Merro, A Calculus of Cyber-Physical Systems, in: LATA, Vol. 10168 of LNCS, Springer, 2017, pp. 115–127. doi:10.1007/978-3-319-53733-7.

[2] Y. Zacchia Lun, A. D'Innocenzo, I. Malavolta, M. D. Di Benedetto, Cyber-Physical Systems Security: a Systematic Mapping Study, CoRR abs/1605.09641.

[3] S. K. Khaitan, J. D. McCalley, Design Techniques and Applications of Cyberphysical Systems: A Survey, IEEE Systems Journal 9 (2) (2015) 350–365.

[4] E. M. Clarke Jr., O. Grumberg, D. A. Peled, Model Checking, MIT Press, 1999.

[5] M. Z. Kwiatkowska, G. Norman, D. Parker, PRISM 4.0: Verification of Probabilistic Real-Time Systems, in: CAV, Vol. 6806 of LNCS, Springer, 2011, pp. 585–591. doi:10.1007/978-3-642-22110-1_47.

[6] R. Milner, The polyadic π-calculus: a tutorial, Tech. Rep. ECS–LFCS–91–180, LFCS (1991).

[7] L. Cardelli, A. Gordon, Mobile Ambients, Theoretical Computer Science 240 (1) (2000) 177–213.

[8] M. Hennessy, J. Riely, A Typed Language for Distributed Mobile Processes, in: POPL, ACM Press, 1998, pp. 378–390. doi:10.1145/268946.268978.

[9] P. J. L. Cuijpers, M. A. Reniers, Hybrid process algebra, Journal of Logic and Algebraic Programming 62 (2) (2005) 191–245.

[10] J. A. Bergstra, C. A. Middleburg, Process Algebra for Hybrid Systems, Theoretical Computer Science 335 (2-3) (2005) 215–280.

[11] D. A. van Beek, K. L. Man, M. A. Reniers, J. E. Rooda, R. R. H. Schiffelers, Syntax and consistent equation semantics of hybrid Chi, The Journal of Logic and Algebraic Programming 68 (1–2) (2006) 129–210.

[12] W. C. Rounds, H. Song, The ϕ-calculus: A language for distributed control of reconfigurable embedded systems, in: HSCC, Vol. 2623 of LNCS, Springer, 2003, pp. 435–449.

[13] V. Galpin, L. Bortolussi, J. Hillston, HYPE: Hybrid modelling by composition of flows, Formal Aspects of Computing 25 (4) (2013) 503–541.

[14] J. Sproston, Decidable model checking of probabilistic hybrid automata, in: FTRTFT, Vol. 1926 of LNCS, Springer, 2000, pp. 31–45. doi:10.1007/3-540-45352-0_5.

[15] J. Hu, J. Lygeros, S. Sastry, Towards a theory of stochastic hybrid systems, in: HSCC, Vol. 1790 of LNCS, Springer, 2000, pp. 160–173.

[16] M. L. Bujorianu, Extended Stochastic Hybrid Systems and their Reachability Problem, in: HSCC, Vol. 2993 of LNCS, Springer, 2004, pp. 234–249. doi:10.1007/978-3-540-24743-2_16.

[17] A. Abate, M. Prandini, J. Lygeros, S. Sastry, Probabilistic Reachability and Safety for Controlled Discrete Time Stochastic Hybrid Systems, Automatica 44 (11) (2008) 2724–2734.

[18] M. Fränzle, E. M. Hahn, H. Hermanns, N. Wolovick, L. Zhang, Measurability and safety verification for stochastic hybrid systems, in: HSCC, ACM, 2011, pp. 43–52. doi:10.1145/1967701.1967710.

[19] E. M. Hahn, A. Hartmanns, H. Hermanns, J. P. Katoen, A Compositional Modelling and Analysis Framework for Stochastic Hybrid Systems, Formal Methods in System Design 43 (2) (2013) 191–232.

[20] S. Wang, N. Zhan, L. Zhang, A Compositional Modelling and Verification Framework for Stochastic Hybrid Systems, Formal Aspects of Computing 29 (4) (2017) 751–775.

[21] M. Hennessy, T. Regan, A Process Algebra for Timed Systems., Information and Computation 117 (2) (1995) 221–239.

[22] R. Segala, Modeling and Verification of Randomized Distributed Real-Time Systems, Ph.D. thesis, MIT (1995).

[23] J. Desharnais, R. Jagadeesan, V. Gupta, P. Panangaden, The Metric Analogue of Weak Bisimulation for Probabilistic Processes, in: LICS, IEEE Computer Society, 2002, pp. 413–422. doi:10.1145/1967701.1967710.

[24] J. Desharnais, V. Gupta, R. Jagadeesan, P. Panangaden, Metrics for labelled Markov processes, Theoretical Computer Science 318 (3) (2004) 323–354.
[25] F. van Breugel, J. Worrell, A behavioural pseudometric for probabilistic transition systems, Theoretical Computer Science 331 (1) (2005) 115–142.
[26] Y. Deng, T. Chothia, C. Palamidessi, J. Pang, Metrics for action-labelled quantitative transition systems, in: QAPL, Vol. 153(2) of ENTCS, 2006, pp. 79–96. doi:10.1016/j.entcs.2005.10.033.
[27] D. Gebler, K. G. Larsen, S. Tini, Compositional Metric Reasoning with Probabilistic Process Calculi, in: FoSSaCS, Vol. 9034 of LNCS, Springer, 2015, pp. 230–245. doi:10.1007/978-3-662-46678-0_15.
[28] D. Gebler, S. Tini, SOS Specifications of Probabilistic Systems by Uniformly Continuous Operators, in: CONCUR, Vol. 42 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015, pp. 155–168. doi:10.4230/LIPIcs.CONCUR.2015.155.
[29] R. Lanotte, M. Merro, S. Tini, Compositional Weak Metrics for Group Key Update, in: MFCS, Vol. 83 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017, pp. 72:1–72:16. doi:10.4230/LIPIcs.MFCS.2017.72.
[30] Y. Deng, R. J. van Glabbeek, M. Hennessy, C. Morgan, Characterising Testing Preorders for Finite Probabilistic Processes, Logical Methods in Computer Science 4 (4) (2008) 1–33.
[31] M. Merro, F. Ballardin, E. Sibilio, A timed calculus for wireless systems, Theoretical Computer Science 412 (47) (2011) 6585–6611.
[32] M. Bernardo, R. De Nicola, M. Loreti, Revisiting trace and testing equivalences for nondeterministic and probabilistic processes, Logical Methods in Computer Science 10 (1). doi:10.2168/LMCS-10(1:16)2014.
[33] A. Cerone, M. Hennessy, M. Merro, Modelling MAC-Layer Communications in Wireless Systems, Logical Methods in Computer Science 11 (1) (2015) 1–59.
[34] R. Lanotte, M. Merro, S. Tini, Weak Simulation Quasimetric in a Gossip Scenario, in: FORTE, Vol. 10321 of LNCS, Springer, 2017, pp. 139–155. doi:10.1007/978-3-319-60225-7_10.
[35] R. Lanotte, M. Merro, S. Tini, Equational Reasonings in Wireless Network Gossip Protocols, CoRR abs/1707.03215.
[36] Y. Deng, W. Du, Logical, metric, and algorithmic characterisations of probabilistic bisimulation, Tech. Rep. CMU-CS-11-110, CMU (March 2011).
[37] C. Villani, Optimal transport: old and new, Springer, 2009.
[38] A. Philippou, I. Lee, O. Sokolsky, Weak Bisimulation for Probabilistic Systems, in: CONCUR, Vol. 1877 of LNCS, 2000, pp. 334–349. doi:10.1007/3-540-44618-4_25.
[39] L. V. Kantorovich, On the transfer of masses, Doklady Akademii Nauk 37 (2) (1942) 227–229, original article in Russian, translation in Management Science, 5 : 1 – 4(1959).
[40] Y. Deng, W. Du, The Kantorovich Metric in Computer Science: A Brief Survey, in: QAPL, Vol. 253(3) of ENTCS, 2009, pp. 73 – 82. doi:10.1016/j.entcs.2009.10.006.
[41] P. Panangaden, Labelled Markov Processes, Imperial College Press, 2009.
[42] F. van Breugel, On behavioural pseudometrics and closure ordinals, Information Processing Letters 112 (19) (2012) 715–718.
[43] D. Gebler, K. G. Larsen, S. Tini, Compositional Bisimulation Metric Reasoning with Probabilistic Process Calculi, Logical Methods in Computer Science 12 (4) (2016) 1–38.
[44] D. Gebler, S. Tini, SOS specifications for uniformly continuous operators, Journal of Computer and System Sciences 92 (2018) 113–151.
[45] K. G. Larsen, A. Skou, Bisimulation through probabilistic testing, Information and Computation 94 (1991) 1–28.
[46] C. Baier, J. P. Katoen, H. Hermanns, B. R. Haverkort, Simulation for Continuous-Time Markov Chains, in: CONCUR, Vol. 2421 of LNCS, Springer, 2002, pp. 338–354. doi:10.1007/3-540-45694-5_23.
[47] C. Baier, H. Hermanns, J. P. Katoen, Probabilistic weak simulation is decidable in polynomial time, Information Processing Letters 89 (3) (2004) 123–130.
[48] M. L. Bujorianu, J. Lygeros, M. C. Bujorianu, Bisimulation for general stochastic hybrid systems, in: HSCC, Vol. 3414 of LNCS, Springer, 2005, pp. 198–214. doi:10.1007/978-3-540-31954-2_13.
[49] T. Chen, S. Kiefer, On the total variation distance of labelled Markov chains, in: CSL-LICS, ACM, 2014, pp. 33:1–33:10. doi:10.1145/2603088.2603099.
[50] A. D'Innocenzo, A. Abate, J. P. Katoen, Robust PCTL model checking, in: HSCC, ACM, 2012, pp. 275–286. doi:10.1145/2185632.2185673.
[51] P. Daca, T. A. Henzinger, J. Kretínský, T. Petrov, Linear distances between Markov chains, in: CONCUR, Vol. 59 of LIPIcs, 2016, pp. 20:1–20:15. doi:10.4230/LIPIcs.CONCUR.2016.20.
[52] H. Wu, F. Noé, Probability distance based compression of hidden Markov models, Multiscale Modeling & Simulation 8 (5) (2010) 1838–1861.
[53] A. Abate, Approximation metrics based on probabilistic bisimulations for general state-space Markov processes: A survey, in: Hybrid Autonomos Systems, Vol. 297 of ENTCS, 2013, pp. 3–25. doi:10.1016/j.entcs.2013.12.002.
[54] G. Bian, A. Abate, On the relationship between bisimulation and trace equivalence in an approximate probabilistic context, in: FoSSaCS, Vol. 10203 of LNCS, Springer, 2017, pp. 321–337. doi:10.1007/978-3-662-54458-7_19.
[55] R. Vigo, F. Nielson, H. Riis Nielson, Broadcast, Denial-of-Service, and Secure Communication, in: IFM, Vol. 7940 of LNCS, Springer, 2013, pp. 412–427. doi:10.1007/978-3-642-38613-8.
[56] I. Lanese, L. Bedogni, M. Di Felice, Internet of Things: a process calculus approach., in: SAC, ACM, 2013, pp. 1339–1346. doi:10.1145/2480362.2480615.
[57] R. Lanotte, M. Merro, A semantic theory of the Internet of Things, Information and Computation 259 (1) (2018) 72–101.
[58] C. Bodei, P. Degano, G. Ferrari, L. Galletta, Tracing where IoT data are collected and aggregated, Logical Methods in Computer Science 13(3) (2017) 1–38.
[59] P. C. Ölveczky, J. Meseguer, Semantics and pragmatics of Real-Time Maude, Higher-Order and Symbolic Computation 20 (1-2) (2007) 161–196.
[60] D. David, K. G. Larsen, A. Legay, M. Mikucionis, Z. Wang, Time for Statistical Model Checking of Real-Time Systems, in: CAV, Vol. 6806 of LNCS, Springer, 2011, pp. 349–355. doi:10.1007/978-3-642-22110-1_27.

[61] L. Benvenuti, D. Bresolin, P. Collins, A. Ferrari, L. Geretti, T. Villa, Ariadne: Dominance Checking of Nonlinear Hybrid Automata Using Reachability Analysis, in: RP, Vol. 7550 of LNCS, Springer, 2012, pp. 79–91. doi:10.1007/978-3-642-33512-9_8.
[62] R. Lanotte, M. Merro, R. Muradore, L. Viganò, A formal approach to cyber-physical attacks, in: CSF, IEEE Computer Society, 2017, pp. 436–450. doi:10.1109/CSF.2017.12.
[63] R. Lanotte, M. Merro, S. Tini, Towards a formal notion of impact metric for cyber-physical attacks, in: iFM, LNCS, Springer, to appear.
[64] R. Lanotte, S. Tini, Weak bisimulation metrics in models with nondeterminism and continuous state spaces, in: ICTAC, LNCS, Springer, to appear.
[65] J. Desharnais, V. Gupta, R. Jagadeesan, P. Panangaden, Approximating labelled Markov processes, Information and Computation 184 (1) (2003) 160–200.