CONFORMANCE RELATIONS AND HYPERPROPERTIES FOR DOPING DETECTION IN TIME AND SPACE

SEBASTIAN BIEWER, RAYNA DIMITROVA, MICHAEL FRIES, MACIEJ GAZDA, THOMAS HEINZE, HOLGER HERMANNS, AND MOHAMMAD REZA MOUSAVI

Saarland University - Computer Science, Saarland Informatics Campus, Saarbrücken, Germany

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Automotive Powertrain Institute, htw saar - University of Applied Sciences, Saarbrücken, Germany

Department of Computer Science, University of Sheffield, Sheffield, UK

Automotive Powertrain Institute, htw saar - University of Applied Sciences, Saarbrücken, Germany

Saarland University - Computer Science, Saarland Informatics Campus, Saarbrücken, Germany and Institute of Intelligent Software, Guangzhou, China

Department of Informatics, King's College London, London, UK

Abstract. We present a novel and generalised notion of doping cleanness for cyber-physical systems that allows for perturbing the inputs and observing the perturbed outputs both in the time– and value–domains. We instantiate our definition using existing notions of conformance for cyber-physical systems. As a formal basis for monitoring conformance-based cleanness, we develop the temporal logic HyperSTL*, an extension of Signal Temporal Logics with trace quantifiers and a freeze operator. We show that our generalised definitions are essential in a data-driven method for doping detection and apply our definitions to a case study concerning diesel emission tests.

1. Introduction

System doping, in our terminology, is an intentional intervention causing a change in the system's normal behaviour against the interests of the user or other stakeholders (such as the society at large). Examples of system doping are widespread and range from vendors' enforcing a monopoly on chargers and spare parts (by checking for and refusing third-party chargers and spare parts, respectively) to tampering with exhaust emission in order to detect and pass emission tests. Doping can be the result of embedding a piece of code or smuggling a piece of electronic circuit into the system and it can be caused by the original developers or by hackers. Software and system doping has been studied in the past couple of years and rigorous theories for it have been developed [8, 18, 9]. These theories were subsequently adopted in order to detect doping, or formally, to check system cleanness [37, 10] (corresponding to the absence of doping).

Key words and phrases: Software Doping, Conformance Testing, Hybrid Systems.

Preprint submitted to
Logical Methods in Computer Science

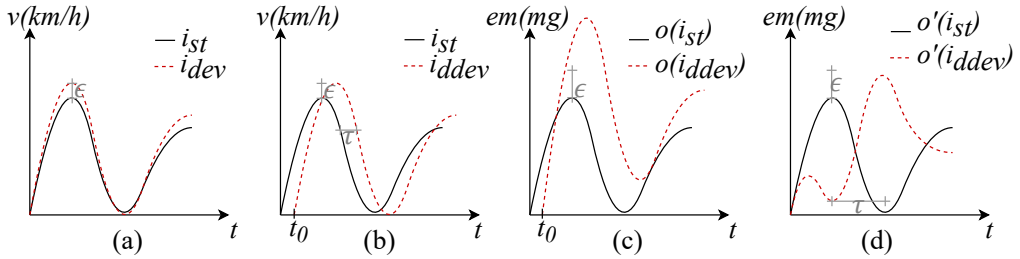

Figure 1: Running Example: Specified (a) and Actual (b) Test Cycles and Emission Footprints obtained from Different (Fictitious) Vehicles (c) and (d).

In the present paper, we extend the theory of doping to the setting of cyber-physical systems (CPS) by exploiting the notions of conformance testing for CPS [2, 21, 39]. The existing theories of software doping define doping in terms of drastic deviations in output as a result of minor deviations in input, where the term "deviation" refers to differences in validity of propositions or values of variables. However, the current notions come short of properly dealing with the issues of retiming and delays, which are commonly present in the signals of CPS. We observe that this is an essential aspect of detecting doping for cyber-physical systems: often the traces to be tested for doping have subtly different timing behaviour, e.g., due to measurement and calibration errors or due to the slight deviations of human actors in acting upon the planned scenarios. The insufficient treatment of retiming and delays can both lead to false negatives, i.e., missing cases of doping, as well as false positives, i.e., reporting spurious doping cases.

To address these issues, we exploit the notion of conformance to devise a general theory of being clean from doping and instantiate that theory with some existing notions of conformance for hybrid systems. We show how these notions can account for retiming and lead to more precise notions of cleanness. Furthermore, we show how the retiming can be synchronized between input and output, leading to a refined notion of cleanness with a rigorous account of the relation between the retiming of input and output.

We illustrate the usefulness of our theory by empirical analysis of diesel engine exhaust emissions in the context of one of the official test cycles, the New European Driving Cycle (NEDC) [48]. In particular, we show that catering for retiming is essential in effectively exploiting the actual driving cycles for performing doping analysis. We thus demonstrate that our new theory remedies a major shortcoming in the existing notions from the literature. To facilitate the presentation, we use throughout the remainder of this paper the following simple running example, which is inspired by our case study.

Example 1.1. Fig. 1.(a) shows two test cycles (evolution of speed over time), designed to detect whether the exhaust emission control of a particular vehicle is doped. The test cycle i_{st}, depicted with a black solid line, is the standard one prescribed by the (fictitious) official regulation, while test cycle i_{dev}, depicted by a red dotted line, is a slight deviation thereof. If the exhaust emissions measured during the test cycle i_{dev} turn out to be significantly higher than the ones measured in test cycle i_{st}, then we can conclude that the exhaust emission system is potentially doped, since it appears tailored to the standard test cycle.

Fig. 1.(b) addresses a notorious problem of testing cars: a human tester is supposed to drive the car as just described, however, she can do this only up to a certain imprecision. Assume her driving of i_{dev} exhibits a slight time shift τ relative to the test cycle, as in i_{ddev}, while i_{st} is being driven as intended.

The result of a test is the emission footprint measured at the exhaust pipe of the car. Fig. 1.(c) and Fig. 1.(d) show two different possible test results (obtained from different cars) for the scenario in Fig. 1.(b). Intuitively, the footprints in Fig. 1.(c) provide significant evidence for doping – a slightly different test cycle has resulted in significantly larger footprint. However, due to the time shift on the input side in Fig. 1.(b) the point-wise difference of the two driven test-cycles has grown very large. As we show in the remainder of this paper, the existing theory of doping fails to detect such a clear evidence, due to the minor delay during the execution of the driving cycle. The emission footprint in Fig. 1.(d) is another (synthetic) example of a significant deviation which cannot be detected for the input in Fig. 1.(b) using existing theories; this latter footprint sheds some light on the intricate design decisions in the theory we develop in this paper.

The contributions for this paper can be summarized as follows:

- We define a *general notion of conformance* that can express different ways of comparing execution traces by allowing deviations both in value and in time;
- We define a general *notion of cleanness for hybrid systems*, and show that it subsumes the existing notion of robust cleanness [18];
- We define the notion of synchronized retiming, which provides a rigorous tool for relating the retiming of input and output and use it to produce a refined notion of cleanness;
- We provide a logical account of cleanness (based on the notion of hybrid conformance) by developing a temporal logic, called HyperSTL⋆, that extends Signal Temporal Logic (STL) with a freeze operator and quantifiers over traces; and
- We demonstrate the usefulness of the proposed generic framework by applying it to *software doping tests* in the automotive domain, where we show that the new cleanness definition is able to flag a case of software doping that goes unnoticed when robust cleanness is used.

This paper substantially extends the theoretical material and the experimental results published in the earlier conference publication [22]. In particular, the following contributions of the present paper are new with respect to the earlier conference publication:

- the notion of cleanness with synchronised retiming in Section 5 and its application in Section 7 for doping detection in our case study,
- the logical approach to cleanness in Section 6, the introduction of our logical formalism HyperSTL⋆, its monitoring, and its application to specify hybrid conformance, and
- the redesign of our experimental setup to comply with the NEDC test cycle (with preconditioning and control of ambient temperature), as well as several new experiments to make full use of our theories of retiming. These new experiments led to much more substantial and decisive evidence for our case study.

2. Related Work

The term "software doping" was coined around 2015 [35] in media uncovering the diesel exhaust emissions scandal. An informal problem formulation [8] pointed out the general phenomenon of intentionally added hidden software behaviour, which is not in the interest of the consumer. Shortly after, this observation has been complemented by a set of formal *cleanness* definitions [18] laying the theoretical foundations upon which formal methods to detect such software behaviour can be used. It is possible to detect missing functionality and undesired existing functionality. The definitions support both sequential programs and nondeterministic reactive programs. To check satisfaction of the definitions, it is necessary

to compare two (or more) execution traces of the same system. Such properties are called *hyperproperties* [16] (whereas classical properties are *trace properties*). Tool support for analysing hyperproperties typically requires high computational effort [15, 30]. There exist several temporal logics for analysing satisfaction of trace properties of various kinds of systems, one of them being *Linear Temporal Logic* (LTL) [44] for systems producing outputs in discrete time steps and properties that do not consider the time passing between outputs. LTL has been extended to the logic HyperLTL, which can express hyperproperties by allowing explicit quantification of execution traces in front of an LTL formula [15]. Tools for model-checking boolean circuits, satisfiability and monitoring of HyperLTL specifications have been developed [4, 13, 30, 25, 26, 27, 28, 34].

Signal Temporal Logic (STL) [42] is an extension of LTL that adds support for time constraints and real-valued signals. Tools exist that automatically try to falsify STL formulas [23, 6]. There has been an extension of STL to HyperSTL in a similar fashion as it was done for HyperLTL [43]. The syntax of HyperSTL, however, is not able to express the cleanness definitions (for deterministic systems) in a way that allows (efficient) falsification. Freeze Temporal Logic [5] introduces a freeze quantifier to "record" the moment of time while evaluating a formula and later use the recorded time to measure time lapse. The concept of freeze quantifier has been used to specify properties of data-dependent systems (using models such as register automata) [20] and dynamical systems [14, 21]. In the present paper, we introduce the concept of freeze quantifier [14] into a logic that is inspired by HyperSTL [43]. The combination turns out to be expressive enough to specify cleanness with respect to hybrid conformance. Moreover, we use a monitoring procedure inspired by an earlier extension of STL with freeze operator [45] to check for cleanness with respect to hybrid conformance. *Robust cleanness* is defined for distance functions on inputs and outputs [18]. When used with temporal logics the distance functions are restricted to those compatible with the logics. To be fully independent, robust cleanness analysis has been embedded into the theory of model-based testing [10] with input-output conformance [46, 47].

Notions of conformance for discrete event systems have been discussed for almost a century. The earliest work on this topic dates back to 1960's when researchers studied model-based testing of digital circuits using Finite State Machine models [36, 41]. Concurrency theory contributed ideas to this field, such as decoupling (i.e., removing the synchronised assumption between) inputs and outputs and observing failures to engage in a communication (and more specifically quiescence) [19, 46]. A theory of conformance testing for systems with continuous dynamics was developed by Michiel van Osch [49]; this theory did not gain much popularity in practice, partly because of its insufficient treatment of approximation (e.g., differences in values and retiming). Pappas and Girard [32, 33] proposed the use of Metric Bisimulation for conformance checking in dynamical systems and Pappas and Fainekos [24] developed a falsification framework for the same purpose. This research led to two notions of conformance used in the present paper, namely hybrid conformance by Abbas and Fainekos [2] and Skorokhod conformance by Deshmukh, Majumdar, and Prabhu [21].

3. Preliminaries

3.1. **Semantic domain.** In this section, we provide definitions regarding semantic domain, conformance, and robust cleanness. We begin with the definition of our semantic domain, called generalised timed traces [31]. This definition subsumes both discrete-time state

sequences and continuous-time trajectories. A generalised timed trace is a function with a discrete or continuous domain (called time domain) and a co-domain which is a metric space. Intuitively, a generalized timed trace maps each element of its time domain to a state. We require that the set of possible states is a metric space since we study conformance notions that compare traces based on the distance between the states of the traces.

Definition 3.1. Let $(\mathcal{Y}, d_{\mathcal{Y}})$ be a metric space. A $\mathcal{Y}$-valued *generalised timed trace (GTT)* is a function $\mu : \mathcal{T} \to \mathcal{Y}$ such that $\mathcal{T} \subseteq \mathbb{R}_{\geq 0}$. We call $\mathcal{T}$ the *time domain* of μ, denoted $dom(\mu)$. $GTT(\mathcal{Y})$ is the set of all $\mathcal{Y}$-valued generalised timed traces.

For a GTT $\mu : \mathcal{T} \to \mathcal{Y}$ and time $t_0 \in \mathcal{T}$, by $\mu[\dots t_0]$ we denote the prefix of μ up to t_0, i.e., the restriction $\mu|_{t \in \mathcal{T} : t \leq t_0}$; likewise, by $\mu[t_s \dots t_e]$, we shall denote the restriction $\mu|_{t \in \mathcal{T} : t_s \leq t \leq t_e}$

A hybrid system is a mapping from generalised (input) timed traces to sets of generalised (output) timed traces.

Definition 3.2. A $\mathcal{Y}$-valued hybrid system is a function $H : GTT(\mathcal{Y}) \to \mathcal{P}(GTT(\mathcal{Y}))$ such that for all $\mu \in GTT(\mathcal{Y})$ and all $\mu' \in H(\mu)$ it holds that $dom(\mu') = dom(\mu)$. We define $\mathcal{H}(\mathcal{Y})$ to be the set of all $\mathcal{Y}$-valued hybrid systems.

In addition, we distinguish deterministic hybrid systems whose output values range over singleton sets only. In what follows, we identify deterministic hybrid systems with functions of the type $GTT(\mathcal{Y}) \to GTT(\mathcal{Y})$.

For simplicity, we assume that the input and output domain are defined on the same metric spaces. The generalisation to different spaces is straightforward.

3.2. **Conformance relations.** Recently, a number of notions of conformance for cyber-physical systems have been proposed [3, 39]. It turns out that these notions, two of which are quoted below, can provide a rigorous basis for doping detection.

Note that throughout the paper, the variables τ and ϵ (with possible subscripts) always range over non-negative real numbers.

Definition 3.3. We say that $\mathcal{Y}$-valued GTTs $\mu_1 : \mathcal{T}_1 \to \mathcal{Y}$ and $\mu_2 : \mathcal{T}_2 \to \mathcal{Y}$ are:

- *trace conformant* with tolerance threshold for signal value ϵ, notation $\mathsf{TraceConf}_\epsilon(\mu_1, \mu_2)$, if $\mathcal{T}_1 = \mathcal{T}_2$ and for all $t \in \mathcal{T}_1$, $d_{\mathcal{Y}}(\mu_1(t), \mu_2(t)) \leq \epsilon$
- *hybrid conformant* with thresholds τ and ϵ, denoted $\mathsf{HybridConf}_{\tau,\epsilon}(\mu_1, \mu_2)$, if:
  - $\forall t_1 \in \mathcal{T}_1 \, \exists t_2 \in \mathcal{T}_2 : |t_2 - t_1| \leq \tau \wedge d_{\mathcal{Y}}(\mu_2(t_2), \mu_1(t_1)) \leq \epsilon$
  - $\forall t_2 \in \mathcal{T}_2 \, \exists t_1 \in \mathcal{T}_1 : |t_1 - t_2| \leq \tau \wedge d_{\mathcal{Y}}(\mu_1(t_1), \mu_2(t_2)) \leq \epsilon$
- *Skorokhod conformant* with tolerance thresholds τ and ϵ, notation $\mathsf{SkorConf}_{\tau,\epsilon}(\mu_1, \mu_2)$, if $\mathcal{T}_1$ and $\mathcal{T}_2$ are intervals and there is a strictly increasing continuous bijection $r : \mathcal{T}_1 \to \mathcal{T}_2$ called retiming, such that:
  - for all $t \in \mathcal{T}_1$, $|r(t) - t| \leq \tau$, and
  - for all $t \in \mathcal{T}_1$, $d_{\mathcal{Y}}(\mu_1(t), \mu_2(r(t))) \leq \epsilon$.

We show in the proposition below and also in our generalisation results in Section 4.2, that these notions are closely related. However, they also have some fundamental differences, that can be illustrated using the example in Fig. 1.

Example 3.4. Consider again the example shown in Fig. 1.

We can see that in Fig. 1.(a) i_{st} and i_{dev} are trace conformant with value threshold ϵ, as they only exhibit point-wise deviations by values less than ϵ. In contrast, i_{st} and i_{ddev} in

Fig. 1.(b) are not trace conformant, yet they are hybrid conformant with time and value margins τ and ϵ, respectively. The key difference is that the inputs depicted in Fig. 1.(b) are very different if compared point-wise, but if one allows for retiming, they are close enough in value after retiming.

The outputs $o'(i_{st})$ and $o'(i_{ddev})$ in Fig. 1.(d) illustrate the fundamental difference between hybrid and Skorokhod conformance: although the order of rising and falling signals are reversed in the two trajectories, they are still hybrid conformant, because hybrid conformance disregards the order. However, Skorokhod conformance requires an order-preserving retiming, and hence distinguishes these two trajectories. On the other hand, such retiming exists, e.g., for i_{st} and i_{ddev} in Fig. 1.(b), witnessing their Skorokhod conformance.

We shall use the following notation. We write $\mathsf{Conf}_1 \sqsubseteq \mathsf{Conf}_2$ whenever for all $\mu_1 : \mathcal{T}_1 \to \mathcal{Y}$ and $\mu_2 : \mathcal{T}_2 \to \mathcal{Y}$, we have $\mathsf{Conf}_1(\mu_1, \mu_2) \implies \mathsf{Conf}_2(\mu_1, \mu_2)$. We write $\mathsf{Conf}_1 \sqsubset \mathsf{Conf}_2$ whenever $\mathsf{Conf}_1 \sqsubseteq \mathsf{Conf}_2$ and $\neg\, \mathsf{Conf}_2 \sqsubseteq \mathsf{Conf}_1$.

Proposition 3.5. *For any $\tau, \epsilon \in \mathbb{R}_{\geq 0}$, the following relations hold:*

$$\mathit{TraceConf}_{\epsilon} \sqsubset \mathit{SkorConf}_{\tau,\epsilon} \sqsubset \mathit{HybridConf}_{\tau,\epsilon}$$

3.3. **Robust cleanness.** We shall now state the original definition of robust cleanness from [18], adapted to our framework of hybrid systems. It is based on Definition 7 and Proposition 19 from [18]; the phrasing below abstracts from the so-called parameters of interest and standard inputs. Moreover it is cast in the setting of generalised timed traces rather than discrete-step programs, and stated using trace conformance with different thresholds for inputs and outputs, κ_I and κ_O.

Intuitively, a hybrid system is robustly clean if for every pair of input prefixes on which no difference in the inputs exceeding κ_I has occurred so far (i.e., all sub-prefixes are trace conformant), the corresponding sets of output prefixes are also conformant with respect to κ_O. As we consider nondeterministic systems, Hausdorff distance is used to compare sets of outputs (see [18] for details).

Definition 3.6. A hybrid system H is robustly clean, denoted $\mathsf{RobustClean}(\kappa_I, \kappa_O)$, whenever:

$$\begin{aligned}
&\forall i_1, i_2 \in GTT(\mathcal{Y}) : \forall t \in dom(i_1) \cup dom(i_2) : \\
&\big(\forall t' \leq t : \mathsf{TraceConf}_{\kappa_I}(i_1[\ldots t'], i_2[\ldots t']) \implies \\
&\quad \big((\forall o_1 \in H(i_1)\, \exists o_2 \in H(i_2) : \mathsf{TraceConf}_{\kappa_O}(o_1[\ldots t], o_2[\ldots t])) \wedge \\
&\quad (\forall o_2 \in H(i_2)\, \exists o_1 \in H(i_1) : \mathsf{TraceConf}_{\kappa_O}(o_1[\ldots t], o_2[\ldots t]))\big)
\end{aligned}$$

Note that in the above definition we do not require that $dom(i_1) = dom(i_2)$. In practice, robust cleanness is typically applied to pairs of traces that are both defined over $\mathbb{N}$. Here, however, for the sake of generality we impose no such restriction. In particular, when the time domains of two traces are different, for example disjoint, the predicate $\mathsf{RobustClean}$ will trivially evaluate to *true*.

Example 3.7. Consider the traces depicted in Fig. 1. The input prefixes i_{st} and i_{ddev} are given in Fig. 1.(b), and the corresponding pair of outputs is shown in Fig. 1.(c). For $t \leq t_0$, we have $i_{ddev}(t) = 0$ and $o(i_{ddev})(t) = 0$. The trace i_{st} results in output $o(i_{st})$ and i_{ddev} results in $o(i_{ddev})$. Suppose that $\epsilon < |i_{st}(t_0) - i_{ddev}(t_0)|$, and for every $t < t_0$ it holds that

$|o(i_{st})(t) - o(i_{ddev})(t)| \leq |i_{st}(t) - i_{ddev}(t)|$. Furthermore, $\epsilon < |o(i_{st})(t_1) - o(i_{ddev})(t_1)|$ at some time $t_1 \geq t_0$. Consider Def. 3.6 instantiated with $\kappa_I = \kappa_O = \epsilon$. Clearly, for every $t < t_0$ the implication in Def. 3.6 is satisfied, since $|o(i_{st})(t') - o(i_{ddev})(t')| \leq |i_{st}(t') - i_{ddev}(t')|$ for all $t' \leq t < t_0$, and $\kappa_I = \kappa_O$. For $t \geq t_0$ the implication is true as well: for $t' < t_0$ the reasoning is as above, and for all $t' \geq t_0$ the left-hand side of the implication is false. Hence, regardless of the difference in the output values at t_1, this pair of inputs satisfies the condition of $\mathsf{RobustClean}(\epsilon, \epsilon)$, and, if these are the only traces in a hybrid system H then we can conclude that H is $\mathsf{RobustClean}(\epsilon, \epsilon)$.

4. Conformance-Based Cleanness

We now define a general notion of conformance-based cleanness and provide two instantiations based on the conformance notions defined in the previous section.

4.1. Motivation.

The need for considering disturbance in time as well as in value is motivated by our running example from Fig. 1. One of the challenges in performing doping tests for cyber-physical systems is that in such systems timing is rarely perfectly precise, due to imprecision in measurements, or caused by the interaction with the physical world. As illustrated in Example 1.1, for instance, when checking for software doping in a car [10], the input to the system is the value of the car's speed over time, which is under the control of a driver, and can thus vary from one execution to the other, even if the driver is trying to execute the same input sequence. Clearly, those variations can be in value, as well as in time.

Example 4.1. Consider the test setup sketched in Fig. 1. There, i_{st} and i_{ddev}, depicted in Fig. 1.(b) define the speed of a car as a function of time. These two input sequences follow a trajectory of values differing by a small margin ϵ (the difference in value allowed by the standard defining the doping tests), but also shifted by a small unit of time τ. Observe further that $|i_{st}(t_0) - i_{ddev}(t_0)| \gg \epsilon$. Thus, without allowing for deviations in time when comparing these input sequences, they will be considered sufficiently different, and as a result their respective exhaust emission outputs will fall out of the comparison when checking for doping according to Def. 3.6, even if the outputs $H(i_{st}(t))$ and $H(i_{ddev}(t))$ are vastly different, as depicted in Fig. 1.(c). This results in a false negative, i.e., failing to detect a clearly doped system.

In the above example, we demonstrated that not accounting for timing disturbances when relating input trajectories can result in false negatives in doping detection. Dually, using the traditional comparison for output traces can result in false positives by requiring overly strict matching of outputs.

The above example motivates the need to account for timing deviations in trajectories. Intuitively, for input trajectories this relaxation results in considering more traces as conforming, and thus enforcing more comparisons when checking if a system is clean. For output trajectories this means relaxing the conformance requirement by considering two output sequences as conforming even if their values are not perfectly aligned in time. Furthermore, different types of timing deviations need to be considered in different scenarios, for example, depending on whether the order in which values occur is important or not.

Example 4.2. Consider the testing workflow from Example 1.1 and Fig. 1, where inputs i_{st} and i_{ddev} are passed to a car. In the second experiment, depicted in Fig. 1.(d), the outputs of the car are $o'(i_{st})$ and $o'(i_{ddev})$, which are hybrid conformant for ϵ and τ. Hence, this observation of the system is classified as clean under hybrid output conformance. However, the output $o'(i_{ddev})$ is clearly suspicious, as the values in $o'(i_{ddev})$ and $o'(i_{st})$ are reversed. This motivates considering conformance notions that require retimings to be order-preserving. Indeed, using Skorokhod conformance we can detect that the system is doped.

The above examples show that in order to be useful in a diverse set of applications, a software cleanness theory should allow for using a variety of conformance notions. To this end, we next take a more general view on conformance notions, in order to be able to develop a generic conformance-based cleanness framework.

4.2. **Retimings and a more generic view on conformance notions.** So far, we have defined three specific notions of conformance which either coincide, or are closely inspired by ones that have appeared in the literature. In order to define a general framework for cleanness, we also wish to treat notions of conformance in a more generic manner. To this end, we propose an abstract definition of conformance predicates. As conformance predicates admit variations in time, as well as in value, our definition is based on *retimings*, a device that will play a key role in the context of this work. In its general form a retiming is a pair of functions between two time domains. Intuitively, given two GTTs, a retiming will define a mapping from points in each of the traces to points in the other trace. Note that in general the mappings are not required to be injective; this way we can cater for notions of conformance allowing for the so-called local disorder phenomenon (in particular hybrid conformance – see Proposition 4.5).

Definition 4.3. A *retiming* is a pair of functions between two time domains, i.e., a pair of the form (r_1, r_2), where $r_1 : \mathcal{T}_1 \to \mathcal{T}_2$ and $r_2 : \mathcal{T}_2 \to \mathcal{T}_1$, with time domains $\mathcal{T}_1, \mathcal{T}_2 \subseteq \mathbb{R}_{\geq 0}$. Given two time domains $\mathcal{T}_1$ and $\mathcal{T}_2$, we denote the set of all retimings between $\mathcal{T}_1$ and $\mathcal{T}_2$ with $\mathcal{RET}(\mathcal{T}_1, \mathcal{T}_2)$.

Retiming is explicitly present in the definition of Skorokhod conformance; there, each Skorokhod retiming is required to be a strictly increasing continuous bijection. We can express a Skorokhod retiming r as an instance of our definition as the pair (r, r^{-1}). In fact, one can also define hybrid conformance, as well as a whole class of conformance notions, using a suitable *family* of retimings.

A family of retimings Ret can be further constrained by τ to a subset Ret_τ of Ret containing only functions that shift time by at most τ time units. In order to use a family of retimings for concrete sequences μ_1 and μ_2, it is necessary to consider only functions that match the domains of the sequences. This leads to a generic notion of conformance associated with a given family of retimings Ret, a given time threshold τ and a given value threshold ϵ.

Definition 4.4. Let Ret be a family of retimings, and let

$$\begin{aligned} \mathsf{Ret}_\tau &\triangleq \{(r_1, r_2) \in \mathsf{Ret} \mid \forall t \in dom(r_i) : \ |r_i(t) - t| \leq \tau \ \ (i = 1, 2)\}, \\ \mathsf{Ret}_\tau(\mathcal{T}_1, \mathcal{T}_2) &\triangleq \mathsf{Ret}_\tau \cap \mathcal{RET}(\mathcal{T}_1, \mathcal{T}_2). \end{aligned}$$

A *conformance notion* with time threshold τ and value threshold ϵ induced by Ret is a predicate $\mathsf{Conf}^{\mathsf{Ret}}_{\tau,\epsilon}$ on pairs of GTTs such that, for $\mu_1 : \mathcal{T}_1 \to \mathcal{Y}$, $\mu_2 : \mathcal{T}_2 \to \mathcal{Y}$:

$$\mathsf{Conf}^{\mathsf{Ret}}_{\tau,\epsilon}(\mu_1,\mu_2) \iff \exists (r_1,r_2) \in \mathsf{Ret}_\tau(\mathcal{T}_1,\mathcal{T}_2) : \quad \forall t \in \mathcal{T}_1 : d_{\mathcal{Y}}(\mu_1(t), \mu_2 \circ r_1(t)) \le \epsilon \\ \wedge \quad \forall t \in \mathcal{T}_2 : d_{\mathcal{Y}}(\mu_2(t), \mu_1 \circ r_2(t)) \le \epsilon.$$

A conformance notion with *unbounded time deviation* and value threshold ϵ induced by Ret is a predicate $\mathsf{Conf}^{\mathsf{Ret}}_{\infty,\epsilon}$ on pairs of GTTs such that, for $\mu_1 : \mathcal{T}_1 \to \mathcal{Y}$, $\mu_2 : \mathcal{T}_2 \to \mathcal{Y}$:

$$\mathsf{Conf}^{\mathsf{Ret}}_{\infty,\epsilon}(\mu_1,\mu_2) \iff \exists (r_1,r_2) \in \mathsf{Ret}(\mathcal{T}_1,\mathcal{T}_2) : \quad \forall t \in \mathcal{T}_1 : d_{\mathcal{Y}}(\mu_1(t), \mu_2 \circ r_1(t)) \le \epsilon \\ \wedge \quad \forall t \in \mathcal{T}_2 : d_{\mathcal{Y}}(\mu_2(t), \mu_1 \circ r_2(t)) \le \epsilon.$$

Unless we state explicitly otherwise, we consider conformance notions with finite time threshold $\tau \in \mathbb{R}_{\ge 0}$. Using the above definition, we can easily express the specific notions of conformance defined in the previous section by selecting a suitable family of retimings.

Proposition 4.5. *The conformance predicates below coincide with the notions of conformance induced by the corresponding families of retimings:*

- $\mathit{TraceConf}_\epsilon$ *is induced by the family of retimings containing only identity functions:* $\mathsf{Ret}_{\mathsf{id}} = \{(\mathsf{id},\mathsf{id}) \mid \mathsf{id} : \mathcal{T} \to \mathcal{T} \text{ is the identity on some } \mathcal{T} \subseteq \mathbb{R}_{\ge 0}\}$.
- $\mathit{SkorConf}_{\tau,\epsilon}$ *is induced by the family of retimings* $\mathsf{Ret} = \{(r, r^{-1}) \mid r \text{ is a strictly increasing continuous bijection}\}$.
- $\mathit{HybridConf}_{\tau,\epsilon}$ *is induced by pairs of arbitrary functions.*

Definition 4.4 also enables us to define other notions of conformance, such as, for instance a "shift conformance", which, intuitively, shifts all time points by a given constant $c \in \mathbb{R}$, i.e., $\mathsf{Ret}_c = \{(r, r^{-1}) \mid r(t) = t + c\}$.

Next, we define a generic notion of cleanness, parametrised by conformance predicates for the input and for the output traces. Instantiating these predicates with existing or new conformance notions, yields different conformance-based notions of cleanness that can capture a variety of cleanness specifications.

4.3. **Definition of Conformance-based Cleanness.** We now extend the notion of robust cleanness [18] to allow for "small" variations in time, in addition to the variations in value. To this end, the new notion makes use of two conformance predicates, one that postulates when two input traces should be considered close enough, and another one that specifies when two output traces are close enough.

Our starting point, the notion of robust cleanness in Definition 3.6, is based on comparison of matching prefixes of a pair of input traces and the corresponding prefixes of the associated output traces. As we now want to accommodate for distance in time, we (1) compare prefixes using a conformance relation, and (2) allow for variation in the length of the compared prefixes that is within the corresponding time-distance threshold. More precisely, when comparing two prefixes, we allow for discarding start and end segments of length at most τ.

This intuition is formalized by the predicate $\mathsf{PrefConf}$ for relaxed comparison of GTT prefixes using a notion of conformance Conf with tolerance threshold τ for time disturbance. We use cascaded notation to define $\mathsf{PrefConf}$ as a higher-order function taking Conf as its first argument. The predicate $\mathsf{PrefConf}$ compares two prefixes μ_1 and μ_2 by requiring that there exist traces $\mu_1[t_1^s \ldots t_1^e]$ and $\mu_2[t_2^s \ldots t_2^e]$ obtained from them, that are conformant with respect to Conf. These traces are obtained by possibly removing a sub-prefix of length at most τ, and/or removing extending with a suffix of length at most τ.

Definition 4.6. Let Conf be a notion of conformance on GTTs with tolerance threshold $\tau \in \mathbb{R}_{\geq 0}$ for time disturbance. For any pair of GTTs $\mu_1 : \mathcal{T}_1 \to \mathcal{Y}$, $\mu_2 : \mathcal{T}_2 \to \mathcal{Y}$, and $t \in \mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$, the predicate $\mathsf{PrefConf}$ is defined as:

$$\begin{aligned}\mathsf{PrefConf}(\mu_1, \mu_2, t) \iff\ & \exists t_1^s \in [0,\tau] \cap \mathcal{T}_1, \exists t_1^e \in [t-\tau, t+\tau] \cap \mathcal{T}_1, \\ & \exists t_2^s \in [0,\tau] \cap \mathcal{T}_2, \exists t_2^e \in [t-\tau, t+\tau] \cap \mathcal{T}_2: \\ & \mathsf{Conf}(\mu_1[t_1^s \ldots t_1^e], \mu_2[t_2^s \ldots t_2^e]).\end{aligned}$$

For conformance notions with unbounded timing deviation $\mathsf{PrefConf}$ coincides with Conf.

The predicate $\mathsf{PrefConf}$ provides a generic notion of prefix-conformance. By instantiating it with conformance relations Conf_I and Conf_O for input and output traces respectively, we define the notion of $(\mathsf{Conf}_I, \mathsf{Conf}_O)$-cleanness.

For deterministic systems $(\mathsf{Conf}_I, \mathsf{Conf}_O)$-cleanness requires that for all pairs of input prefixes for which all sub-prefixes are prefix-conformant w.r.t. Conf_I, the corresponding pair of output prefixes are prefix-conformant w.r.t. Conf_O.

Definition 4.7. A deterministic system H is $(\mathsf{Conf}_I, \mathsf{Conf}_O)$-clean if

$$\begin{aligned}\forall i_1, i_2 \in\ & GTT(\mathcal{Y}) : \forall t \in dom(i_1) \cup dom(i_2) : \\ & \big(\forall t' \leq t :\ \mathsf{PrefConf}_I(i_1, i_2, t')\big) \implies \mathsf{PrefConf}_O(H(i_1), H(i_2), t).\end{aligned}$$

The above definition naturally generalises to nondeterministic hybrid systems, by comparing sets of possible output prefixes using Hausdorff distance as in [18].

Definition 4.8. A system H is $(\mathsf{Conf}_I, \mathsf{Conf}_O)$-clean if

$$\begin{aligned}& \forall i_1, i_2 \in GTT(\mathcal{Y}) : \forall t \in dom(i_1) \cup dom(i_2) : \\ & \big(\forall t' \leq t :\ \mathsf{PrefConf}_I(i_1, i_2, t')\big) \implies \\ & \qquad \big((\forall o_1 \in H(i_1)\, \exists o_2 \in H(i_2) :\ \mathsf{PrefConf}_O(o_1, o_2, t)) \wedge \\ & \qquad\ (\forall o_2 \in H(i_2)\, \exists o_1 \in H(i_1) :\ \mathsf{PrefConf}_O(o_1, o_2, t))\big).\end{aligned}$$

Robust cleanness [18] can be now formulated as conformance-based cleanness, which establishes that $(\mathsf{Conf}_I, \mathsf{Conf}_O)$-cleanness is a generalisation. Using hybrid conformance, we define hybrid-conformance cleanness, and similarly, plugging in Skorokhod conformance, we define Skorokhod-conformance cleanness. Formally:

- A hybrid system H is robustly clean, denoted $\mathsf{RobustClean}(\kappa_I, \kappa_O)$, if and only if H is $(\mathsf{TraceConf}_{\kappa_I}, \mathsf{TraceConf}_{\kappa_O})$-clean.
- A hybrid system H is *hybrid-conformance clean with conformance thresholds* $(\tau_I, \epsilon_I, \tau_O, \epsilon_O)$, which we denote by $\mathsf{HybridClean}(\tau_I, \epsilon_I, \tau_O, \epsilon_O)$, if and only if H is $(\mathsf{HybridConf}_{\tau_I,\epsilon_I}, \mathsf{HybridConf}_{\tau_O,\epsilon_O})$-clean.
- A hybrid system H is *Skorokhod-conformance clean with conformance thresholds* $(\tau_I, \epsilon_I, \tau_O, \epsilon_O)$, denoted $\mathsf{SkorClean}(\tau_I, \epsilon_I, \tau_O, \epsilon_O)$, if and only if H is $(\mathsf{SkorConf}_{\tau_I,\epsilon_I}, \mathsf{SkorConf}_{\tau_O,\epsilon_O})$-clean.

4.4. **Properties.** We will now establish some key relations between the cleanness notions defined previously. We begin by lifting the implication between conformance relations to implication between cleanness notions defined using those relations.

Proposition 4.9. *Suppose that* $\mathsf{Conf}_I^1 \sqsupseteq \mathsf{Conf}_I^2$ *and* $\mathsf{Conf}_O^1 \sqsubseteq \mathsf{Conf}_O^2$. *Then for any system* H: H *is* $(\mathsf{Conf}_I^1, \mathsf{Conf}_O^1)$*-clean* $\implies$ H *is* $(\mathsf{Conf}_I^2, \mathsf{Conf}_O^2)$*-clean.*

The proposition above has two important corollaries. The first one explains the relationships between the original robust cleanness, and notions of cleanness based on Skorokhod conformance and hybrid conformance, in particular stating the conservative generalisation property for the latter notions. The second corollary compares cleanness notions with different conformance thresholds.

Corollary 4.10. *For all $\tau_I, \tau_O, \epsilon_I, \epsilon_O \in \mathbb{R}_{\geq 0}$, the following implications hold:*
(1) $\mathit{RobustClean}(\epsilon_I, \epsilon_O) \implies \mathit{SkorClean}(0, \epsilon_I, \tau_O, \epsilon_O) \implies \mathit{HybridClean}(0, \epsilon_I, \tau_O, \epsilon_O)$,
(2) $\mathit{HybridClean}(\tau_I, \epsilon_I, 0, \epsilon_O) \implies \mathit{SkorClean}(\tau_I, \epsilon_I, 0, \epsilon_O) \implies \mathit{RobustClean}(\epsilon_I, \epsilon_O)$.
Also, $\mathit{RobustClean}(\epsilon_I, \epsilon_O) = \mathit{SkorClean}(0, \epsilon_I, 0, \epsilon_O) = \mathit{HybridClean}(0, \epsilon_I, 0, \epsilon_O)$ and hence SkorClean and HybridClean are conservative extensions of robust cleanness.

Corollary 4.11. *For all $\epsilon_I, \epsilon'_I, \epsilon_O, \epsilon'_O, \tau_I, \tau'_I, \tau_O, \tau'_O$ that satisfy the inequalities $\epsilon'_I \leq \epsilon_I, \quad \tau'_I \leq \tau_I, \quad \epsilon'_O \geq \epsilon_O, \quad \tau'_O \geq \tau_O$ the following implications hold:*
(1) $\mathit{RobustClean}(\epsilon_I, \epsilon_O) \implies \mathit{RobustClean}(\epsilon'_I, \epsilon'_O)$;
(2) $\mathit{HybridClean}(\epsilon_I, \tau_I, \epsilon_O, \tau_O) \implies \mathit{HybridClean}(\epsilon'_I, \tau'_I, \epsilon'_O, \tau'_O)$;
(3) $\mathit{SkorClean}(\epsilon_I, \tau_I, \epsilon_O, \tau_O) \implies \mathit{SkorClean}(\epsilon'_I, \tau'_I, \epsilon'_O, \tau'_O)$.

Example 4.12. Consider the testing workflow in Fig. 1. The inputs passed to a car are i_{st} and i_{ddev}, depicted in Fig. 1.(b). One of the test results is presented in Fig. 1.(c), where i_{st} reveals output $o(i_{st})$ and i_{ddev} reveals $o(i_{ddev})$. We assume that $\epsilon < |i_{st}(t_0) - i_{ddev}(t_0)|$ and $\epsilon < |o(i_{st})(t_1) - o(i_{ddev})(t_1)|$ at some time $t_1 \geq t_0$.

- As we saw in Example 3.7, for inputs i_{st} and i_{ddev}, the car that emits the outputs depicted in Fig. 1.(c) is deemed $\mathsf{RobustClean}(\epsilon, \epsilon)$. Note, that in the presence of other inputs the car used for testing might not be $\mathsf{RobustClean}(\epsilon, \epsilon)$.
- As explained in Example 3.4, i_{st} and i_{ddev} are hybrid conformant for ϵ and τ, i.e., the predicate $\mathsf{PrefConf}_I$ on the left-hand side of the implication in Def. 4.7 holds. $\mathsf{PrefConf}_O$, however, fails at time t_1 for signals $o(i_{st})$ and $o(i_{ddev})$. Hence, the system tested in Fig. 1.(c) is not $\mathsf{HybridClean}(\epsilon, \tau, \epsilon, \tau)$.

We now discuss testing and falsification of conformance-based cleanness. For systems with discrete time domains the existing methods for verifying [18] or testing [10] robust cleanness can be readily applied.

In the case of hybrid cleanness, existing methods for testing hybrid conformance, such as [1] and [7] can be extended to testing and falsification of hybrid cleanness of hybrid systems consisting of traces with finite time domains. Methods for checking Skorokhod conformance were presented in [21]. Due to the quantification over all time-points t' in our Definition 4.7 and Definition 4.8, it is not clear how to directly extend them to testing Skorokhod cleanness.

5. Cleanness with Synchronized Retimings

5.1. **Practical motivation.** An intuitive and useful notion of doping cleanness should capture precisely what we expect from a clean system subject to disturbances in time and value. In this regard, one can observe that even the more discriminating SkorClean predicate has certain drawbacks. The following example motivates why one may want to resort to the finer definition to be proposed in this section.

Example 5.1. Consider the scenario of particle emission cleanness presented in Example 3.4 and the input (velocity)- and output trajectories depicted in Fig. 2. Assume that for some input trajectory i_1, the vehicle shows the output (emission) profile o_1; for a second input $i_2(t) = i_1(t - \tau)$, consider two possible output trajectories: one output is $o_1(t - \tau)$, i.e., it is shifted in the same manner as input; this is assumed to be the best response to i_2. The other output is of the form $o_1(t - \tau - \delta)$, where $\delta > 0$ can be arbitrarily small, i.e., it is the optimal output with an arbitrary small shift to the right. Skorokhod-conformance cleanness with $\tau_I = \tau_O$ would accept the first output, but it would reject the second one. A potential solution could be to increase the value of τ_O so that it is significantly larger than τ_I, but this increases the imprecision by accepting too many trajectories shifted far to the left from $o_1(t - \tau)$.

Intuitively, when the input shifts by some τ, we would like to compare the corresponding output trajectory with the one that is shifted accordingly. In the above-mentioned case, one would therefore ideally like to perform conformance check of output against $o_1(t - \tau)$, rather than $o_1(t)$.

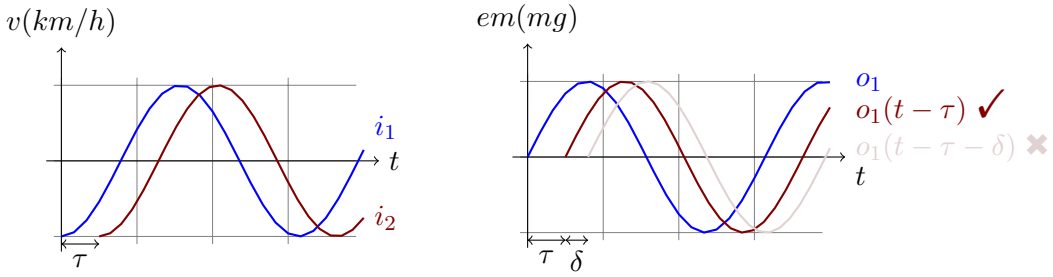

Figure 2: Imprecision problem in Skorokhod-conformance cleanness without synchronisation of retimings. While $o_1(t - \tau)$ is the best expected response to i_2, no trajectory to the right of $o_1(t - \tau)$ is accepted when $\tau_I = \tau_O = \tau$.

5.2. **Formal theory of synchronized retiming.** In order to alleviate this imprecision, we propose a definition of conformance-based cleanness with synchronised retimings, in which we do not check the conformance of the resulting outputs directly, but rather check conformance of each of the outputs against the transformation of the other output with the retiming that is expected, based on the retiming of the corresponding input. Note that the expected retiming of the output is not always precisely the same as that of the input. Instead, we assume that the set of expected output retimings to a given input retiming is available through a synchronisation function.

As mentioned earlier, we can include in the set of conforming output trajectories the best expected response $o_1(t - \tau)$ by allowing a sufficiently large τ_O, but this comes at the price of introducing imprecision in the conformance relation. By shifting the reference point of conformance comparison, our cleanness with synchronised retimings avoids this imprecision. What is more important, by performing the synchronisation independently of τ_O, we introduce the opportunity to constrain the set of conforming output traces to those traces that are as close as desired to the ideal expected output behaviour.

We proceed to formalise the enhanced notion of cleanness with synchronisation outlined above. We start with two auxiliary definitions.

Definition 4.4 entails that whenever the conformance predicate Conf holds for certain pair of timed traces, it is witnessed by at least one relevant retiming (r_1, r_2). The following operator "extracts" all such witness retimings:

$$\begin{aligned}\mathsf{Wit\ Conf}^{\mathsf{Ret}}_{\tau,\epsilon}(\mu_1,\mu_2) \triangleq\ & \{(r_1,r_2) \in \mathsf{Ret}_\tau(\mathcal{T}_1,\mathcal{T}_2) \mid \\ & \forall t \in \mathcal{T}_1 : d_{\mathcal{Y}}(\mu_1(t) - \mu_2 \circ r_1(t)) \leq \epsilon \wedge \\ & \forall t \in \mathcal{T}_2 : d_{\mathcal{Y}}(\mu_1 \circ r_2(t) - \mu_2(t)) \leq \epsilon\}.\end{aligned}$$

Similarly, we define the collection of all retimings that witness prefix conformance ($\mathsf{PrefConf}$ predicate, Definition 4.6):

$$\begin{aligned}\mathsf{PrefixWit\ Conf}(\mu_1,\mu_2,t) \triangleq\ & \{(r_1,r_2) \in \mathsf{Wit\ Conf}(\mu_1[t_1^s \ldots t_1^e], \mu_2[t_2^s \ldots t_2^e]) \mid \\ & t_1^s \in [0,\tau] \cap \mathcal{T}_1, t_1^e \in [t-\tau, t+\tau] \cap \mathcal{T}_1, \\ & t_2^s \in [0,\tau] \cap \mathcal{T}_2, t_2^e \in [t-\tau, t+\tau] \cap \mathcal{T}_2\}.\end{aligned}$$

Note that the domains of the retimings in $\mathsf{PrefixWit\ Conf}(\mu_1,\mu_2,t)$ can be smaller than the domains of μ_1 and μ_2.

Synchronisation is realised through a function Sync specifying all allowed pairs of output retimings for a given pair of input retimings. With this, we can extend the definition of $(\mathsf{Conf}_I, \mathsf{Conf}_O)$ cleanness as follows. Given two inputs that are conformant, i.e., $\mathsf{PrefConf}_I(i_1,i_2)$, we may pick any pair (r_1,r_2) from the set $\mathsf{PrefixWit\ Conf}_I(i_1,i_2)$ of pairs of retiming functions for which the input conformance holds. This pair induces another pair $(r_1',r_2') \in \mathsf{Sync}(r_1,r_2)$ of retiming functions for the output timeline. For those, the prefix conformance predicates $\mathsf{Conf}_O(o_1 \circ r_2', o_2)$ and $\mathsf{Conf}_O(o_1, o_2 \circ r_1')$ must hold. This is formally expressed in the following definition.

Definition 5.2. A deterministic system H is $(\mathsf{Conf}_I, \mathsf{Conf}_O)$*–clean with synchronised retiming through* $\mathsf{Sync} : \mathcal{RET}(\mathcal{T}_1,\mathcal{T}_2) \to \mathcal{P}(\mathcal{RET}(\mathcal{T}_1,\mathcal{T}_2))$ if the following holds:

$$\begin{aligned}\forall i_1,i_2 \in GTT(\mathcal{Y}) : \quad & \forall t \in dom(i_1) \cup dom(i_2) \\ & \big(\forall t' \leq t : \mathsf{PrefConf}_I(i_1,i_2,t')\big) \implies \\ & \exists (r_1,r_2) \in \mathsf{PrefixWit\ Conf}_I(i_1,i_2,t) : \\ & \exists (r_1',r_2') \in \mathsf{Sync}(r_1,r_2) : \\ & \mathsf{PrefConf}_O(H(i_1) \circ r_2', H(i_2), t) \wedge \\ & \mathsf{PrefConf}_O(H(i_1), H(i_2) \circ r_1', t)\end{aligned}$$

Through the function Sync, which is a parameter to the above definition, we can specify the allowed retimings for the output, such as, for example a scaling of the input retiming when the timelines of the input and the output have different scales. It is the responsibility of the cleanness tester or verifier to accurately specify the expected behaviour, as an inappropriately chosen Sync function can result in declaring doped systems to be clean.

One important aspect of this definition is that by selecting a suitable synchronisation function Sync we can incorporate in the cleanness check any available knowledge regarding the expected output behaviours for conforming input trajectories. The following proposition states how the conformance-based notions of cleanness can be recovered by choosing appropriate retimings.

Proposition 5.3. *For a given class of retimings* Ret*, an arbitrary induced conformance relation on inputs* $\mathsf{Conf}_I = \mathsf{Conf}^{\mathsf{Ret}}_{\tau,\epsilon}$*, and a conformance relation on outputs* Conf_O*, there*

exists a synchronised retiming **Sync** *such that* $(\mathsf{Conf}_I, \mathsf{Conf}_O)$*-cleanness (as per Definition 4.7) is a special instance of* $(\mathsf{Conf}_I, \mathsf{Conf}_O)$*-cleanness through* **Sync**.

By setting $\mathsf{Sync}(r_1, r_2) = \{(\mathsf{id}, \mathsf{id})\}$ we obtain the corresponding notion of $(\mathsf{Conf}_I, \mathsf{Conf}_O)$–cleanness, which, in particular, means that cleanness with synchronised retimings is also a conservative generalization of robust cleanness.

Example 5.4. Consider the behaviour introduced in Example 5.1. As for the retiming witnessing conformance between i_1 and i_2, let us take the most obvious one i.e. $(r_1, r_2) = (t - \tau, t + \tau)$. If (r_1, r_2) covers the whole domain of the output trace, then we can use $\mathsf{Sync}(r_1, r_2) = \{(r_1, r_2)\}$, according to which the output should be retimed in the same way as the input is. By reusing the same retiming of input for output, $o_1(t - \tau - \delta)$ conforms to the retimed output o_1 with respect to the margin δ.

We use this theory in our experimental setup in Section 7 and show how it can lead to a more accurate analysis of emission data in practice.

6. Expressing Cleanness in Timed Hyper Logics

In this section we introduce a logic that is capable of characterizing the notions of robust and hybrid cleanness. Since robust cleanness can be characterized in the logic HyperLTL [15], the logic we propose is a temporal logic for hyperproperties. Our semantic domain consists of generalized timed traces, and thus, our logic extends Signal Temporal Logic (STL) [42] with quantifiers over traces. In order to be able to express deviations in time, our logic uses freeze quantifiers as the mechanism for comparing values at different time points. More precisely, the proposed logic is obtained by extending STL* [14] with trace quantifiers. In the remainder of the section we provide the formal definition of the logic and discuss its applicability in the context of specifying and monitoring cleanness of hybrid systems.

6.1. Preliminaries.

For the presentation in this section it will be convenient to consider hybrid systems as sets of GTTs, where each GTT represents a pair of input and output GTTs. The reason for this is that we will define a logic whose formulas refer to both the inputs and the outputs of a hybrid system over time, and are therefore interpreted over sets of such combined traces that contain both the input to the system and the system's output.

Formally, we will represent a $\mathcal{Y}'$-valued hybrid system $H : GTT(\mathcal{Y}') \to \mathcal{P}(GTT(\mathcal{Y}'))$ as a subset of $GTT(\mathcal{Y})$ where $\mathcal{Y} = \mathcal{Y}' \times \mathcal{Y}'$, defined as $\{\mu \in GTT(\mathcal{Y}' \times \mathcal{Y}') \mid \exists \mu_I, \mu_O \in GTT(\mathcal{Y}'), \mu_O \in H(\mu_I), \mu(t) = (\mu_I(t), \mu_O(t)) \text{ for all } t \in dom(\mu_I)\}$. The definition of the GTTs μ in this set is possible since according to Definition 3.2 we have that for all $\mu_I \in GTT(\mathcal{Y}')$ and all $\mu_O \in H(\mu_I)$ it holds that $dom(\mu_I) = dom(\mu_O)$.

For the rest of this subsection, whenever we refer to a GTT $\mu \in H$ of a $\mathcal{Y}'$-valued hybrid system H, we mean a function $\mu : \mathcal{T} \to \mathcal{Y}$ defined as above, with $\mathcal{Y} = \mathcal{Y}' \times \mathcal{Y}'$. Given $\mu \in H$ such that $\mu(t) = (\mu^I(t), \mu^O(t))$ for $t \in dom(\mu)$, we denote with $\mu^I : \mathcal{T} \to \mathcal{Y}'$ the projection of μ on the input component and with $\mu^O : \mathcal{T} \to \mathcal{Y}'$ its projection on the output component.

Let $\mu : \mathcal{T} \to \mathcal{Y}$ be a GTT. If $\mathcal{T}$ is an interval of the form $[0, b] \subseteq \mathbb{R}_{\geq 0}$ with $b \in \mathbb{Q}_{>0}$, or of the form $[0, \infty)$, and $\mathcal{Y} = \mathbb{R}^n$ for some $n \in \mathbb{N}$, we say that μ is a *real-valued signal*, and define $length(\mu) = b$, respectively $length(\mu) = \infty$, to be the *time length* of μ. If $\mathcal{T}$ is instead a strictly increasing sequence $t_0, t_1, t_2, \ldots$ of rational numbers such that $t_0 = 0$ we say that μ is a *timed word* and similarly define its *time length* as $length(\mu) = \max \mathcal{T}$ if $\mathcal{T}$ is finite and $length(\mu) = \infty$ otherwise.

Let X be a finite set of *real-valued variables*. We denote with $\mathbb{R}^X$ the set of possible valuations of X. In the rest of this section we assume that the range of all GTTs that we consider is $\mathbb{R}^X$ for a given finite set of real-valued variables. We will assume that the variables in X are indexed, i.e., $X = \{x_1, x_2, \dots x_n\}$ for some $n \in \mathbb{N}$, and use $\mathbb{R}^n$ instead of $\mathbb{R}^X$ with the expected interpretation. An *atomic predicate* over X is a function $\alpha : \mathbb{R}^X \to \mathbb{B}$. Recall from Definition 3.1 that a GTT $\mu : \mathcal{T} \to \mathcal{Y}$ is defined for a metric space $(\mathcal{Y}, d_{\mathcal{Y}})$. When $\mathcal{Y} = \mathbb{R}^X$ for some set of variables X, we assume that the metric d can be expressed as an arithmetic expression $d_{\mathcal{Y}}(X, X')$ over the variables $X \cup X'$, where $X' = \{x' \mid x \in X\}$. More precisely, we have that the expression $d_{\mathcal{Y}}(X, X')$ evaluates to $u \in \mathbb{R}$ for valuations $v \in \mathbb{R}^X$ and $v' \in \mathbb{R}^{X'}$ of X and X', respectively, if and only if $d_{\mathcal{Y}}(v, v') = u$. For a $\mathcal{Y}'$-valued hybrid system H we denote with d_I and d_O the arithmetic expressions that define the metrics associated with the underlying metric spaces for the input and output values of H.

6.2. **The logic HyperSTL*.** We now define the logic HyperSTL*, which extends the logic STL* [14] with *quantifiers over traces*, that are used to relate multiple GTTs in a hybrid system. To this end, let V_{trace} be a countably infinite set of *trace variables*. For a set X of real-valued variables and a given trace variable $\pi \in V_{trace}$, let $X_\pi = \{x_\pi \mid x \in X\}$ be the set of variables indexed with π.

Let $\mathcal{I} = \{1, \dots, m\}$ for some $m \in \mathbb{N}$ be a finite index set. As in the logic STL*, the index set $\mathcal{I}$ consists of the indices of the positions in the *frozen time vector*. Intuitively, at each position of the frozen time vector a time point can be stored. For a trace variable $\pi \in V_{trace}$ and an index $i \in \mathcal{I}$, let $X_\pi^{*_i} = \{x_\pi^{*_i} \mid x \in X\}$ be the set of variables indexed with π and $*_i$.

6.2.1. *Syntax.* Let X be a finite set of real-valued variables, and AP be a set of atomic predicates over the set of indexed variables $\bigcup_{\pi \in V_{trace}} X_\pi \cup \bigcup_{\pi \in V_{trace}, i \in \mathcal{I}} X_\pi^{*_i}$.

HyperSTL* formulas are defined according to the following grammar.

$$\begin{aligned} \Phi &::= \exists \pi.\Phi \mid \forall \pi.\Phi \mid \varphi, \\ \varphi &::= \alpha \mid \top \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \,\mathcal{U}_J\, \varphi \mid \varphi \,\mathcal{S}_J\, \varphi \mid *_i \varphi, \end{aligned}$$

where $\pi \in V_{trace}$ is a trace variable, α is an atomic predicate from AP, $J \subseteq \mathbb{R}_{\geq 0}$ is an interval with endpoints in $\mathbb{Q}_{\geq 0} \cup \{\infty\}$, and $i \in \mathcal{I}$ is an index.

The operators $\mathcal{U}$ and $\mathcal{S}$ are the temporal operators *Until* and *Since*. The $*_i$ operator, for $i \in \mathcal{I}$ is the *signal-value freeze* operator. Their semantics is formally defined below. When the interval J is of the form $[0, \infty)$ we often omit it for convenience.

The Boolean constant $\bot$ (false), additional Boolean operators, as well as additional temporal operators are defined in the usual way. More concretely, we define $\Diamond_J \varphi \stackrel{\triangle}{=} \top \,\mathcal{U}_J\, \varphi$, $\Box_J \varphi \stackrel{\triangle}{=} \neg \Diamond_J \neg\varphi$, $\Diamondblack_J \varphi \stackrel{\triangle}{=} \top \,\mathcal{S}_J\, \varphi$, and $\boxminus_J \varphi \stackrel{\triangle}{=} \neg \Diamondblack_J \neg\varphi$.

A HyperSTL* formula is *well-formed* if each occurrence of a trace quantifier introduces a unique variable name, and it is *closed* if every occurrence of a variable in X_π is in the scope of a quantifier for π. We will consider only well-formed HyperSTL* formulas.

Note that in HyperSTL*, unlike [14], we also allow the past operator $\mathcal{S}$, as well as arbitrary intervals J in the operators $\mathcal{U}$ and $\mathcal{S}$. We define HyperSTL*$_{fin}$ to be the fragment of HyperSTL* such that every interval J is of the form $[a, b]$, where $a, b \in \mathbb{Q}_{\geq 0}$ and $a < b$.

6.2.2. *Semantics.* HyperSTL* formulas are interpreted over trace assignments and register valuations. A *trace assignment* is a partial function with finite domain from V_{trace} to the set of GTTs in a given hybrid system. Formally, given a hybrid system H represented as a set of input-output GTTs, a trace assignment Π is a partial function $\Pi : V_{trace} \to H$. *Register valuations* are $|\mathcal{I}|$-dimensional vectors over $\mathbb{R}_{\geq 0}$.

Let Π be a trace assignment with domain $U_{trace} \subseteq V_{trace}$, and let α be an atomic predicate defined over the variables in $\bigcup_{\pi \in U_{trace}} X_\pi \cup \bigcup_{\pi \in U_{trace}, i \in \mathcal{I}} X_\pi^{*i}$. Consider a time point $t \in \mathbb{R}_{\geq 0}$ such that for each $\pi \in V_{trace}$ for which a variable from X_π occurs in α it holds that $t \in dom(\Pi(\pi))$, and a register valuation T such that for each pair $\pi \in V_{trace}$ and $i \in \mathcal{I}$ such that a variable from X_π^{*i} occurs in α it holds that $T(i) \in dom(\Pi(\pi))$. Then, the value of the atomic predicate α at the tuple (Π, T, t) is defined as:

$$\alpha(\Pi, T, t) \stackrel{\triangle}{=} \alpha((\Pi(\pi)(t))_{\pi \in U_{trace}}, (\Pi(\pi)(T(i)))_{\pi \in U_{trace}, i \in \mathcal{I}}).$$

Intuitively, the atomic predicate is evaluated using the signal values at time point t and at the time points stored in the frozen time vector T. If for some of the indexed variables that occur in α the corresponding time point is not in the time domain of the corresponding trace, then the value of the atomic predicate is *undefined*.

To define the semantics of HyperSTL*, we define the function Value that maps a formula Ψ, a trace assignment Π, a register assignment T and a time point t to a value in the set $\{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$, which indicates whether Ψ is true (T), false (F) or undefined (U) at (Π, T, t). Formally, for a hybrid system H, a trace assignment Π, a register valuation T, and $t \in \mathbb{R}_{\geq 0}$, the value $\mathsf{Value}(\Psi, H, \Pi, T, t)$ is defined by induction on the structure of HyperSTL* formulas.

- If $\Psi = \alpha$, then
$$\mathsf{Value}(\Psi, H, \Pi, T, t) = \begin{cases} \alpha(\Pi, T, t) & \text{if the value of } \alpha \text{ is defined at } (\Pi, T, t), \\ \mathsf{U} & \text{otherwise.} \end{cases}$$
- If $\Psi = \top$, then $\mathsf{Value}(\Psi, H, \Pi, T, t) = \mathsf{T}$.
- If $\Psi = \neg\varphi$, then
$$\mathsf{Value}(\Psi, H, \Pi, T, t) = \begin{cases} \mathsf{T} & \text{if } \mathsf{Value}(\varphi, H, \Pi, T, t) = \mathsf{F}, \\ \mathsf{F} & \text{if } \mathsf{Value}(\varphi, H, \Pi, T, t) = \mathsf{T}, \\ \mathsf{U} & \text{if } \mathsf{Value}(\varphi, H, \Pi, T, t) = \mathsf{U}. \end{cases}$$
- If $\Psi = \varphi_1 \vee \varphi_2$, then
$$\mathsf{Value}(\Psi, H, \Pi, T, t) = \begin{cases} \mathsf{T} & \text{if } \mathsf{Value}(\varphi_1, H, \Pi, T, t) = \mathsf{T} \text{ or } \mathsf{Value}(\varphi_2, H, \Pi, T, t) = \mathsf{T}, \\ \mathsf{F} & \text{if } \mathsf{Value}(\varphi_1, H, \Pi, T, t) = \mathsf{F} \text{ and } \mathsf{Value}(\varphi_2, H, \Pi, T, t) = \mathsf{F}, \\ \mathsf{U} & \text{otherwise.} \end{cases}$$
- If $\Psi = \varphi_1 \mathcal{U}_J \varphi_2$, then
$$\mathsf{Value}(\Psi, H, \Pi, T, t) = \begin{cases} \mathsf{T} & \text{if for some } t' \geq t \text{ such that } t' - t \in J: \ \mathsf{Value}(\varphi_2, H, \Pi, T, t') = \mathsf{T} \text{ and} \\ & \quad \text{for all } t'' \in [t, t'): \ \mathsf{Value}(\varphi_1, H, \Pi, T, t'') \in \{\mathsf{T}, \mathsf{U}\} \text{ or} \\ & \qquad t'' - t \in J \text{ and } \mathsf{Value}(\varphi_2, H, \Pi, T, t'') = \mathsf{T}, \\ \mathsf{F} & \text{otherwise.} \end{cases}$$

- If $\Psi = \varphi_1 \, \mathcal{S}_J \, \varphi_2$, then

$$\mathsf{Value}(\Psi, H, \Pi, T, t) = \begin{cases} \mathsf{T} & \text{if for some } t' \in [0,t] \text{ such that } t - t' \in J: \ \mathsf{Value}(\varphi_2, H, \Pi, T, t') = \mathsf{T} \text{ and} \\ & \quad \text{for all } t'' \in (t', t]: \ \mathsf{Value}(\varphi_1, H, \Pi, T, t'') \in \{\mathsf{T}, \mathsf{U}\} \text{ or} \\ & \qquad\qquad t'' - t \in J \text{ and } \mathsf{Value}(\varphi_2, H, \Pi, T, t'') = \mathsf{T}, \\ \mathsf{F} & \text{otherwise.} \end{cases}$$

- If $\Psi = *_i \varphi$, then

$$\mathsf{Value}(\Psi, H, \Pi, T, t) = \begin{cases} \mathsf{Value}(\varphi, H, \Pi, T[i \mapsto t], t) & \text{if } t \in \mathit{dom}(\Pi(\pi)) \text{ for all } x_\pi^{*i} \text{ that appear in } \varphi, \\ \mathsf{U} & \text{otherwise.} \end{cases}$$

  where $T[i \mapsto t](j) \stackrel{\triangle}{=} t$ if $j = i$ and $T[i \mapsto t](j) \stackrel{\triangle}{=} T(j)$ if $j \neq i$.
- If $\Psi = \exists \pi . \Phi$, then

$$\mathsf{Value}(\Psi, H, \Pi, T, t) = \begin{cases} \mathsf{T} & \text{if for some } \mu \in H: \ \mathsf{Value}(\Phi, H, \Pi[\pi \mapsto \mu], T, t) = \mathsf{T}, \\ \mathsf{F} & \text{otherwise.} \end{cases}$$

- If $\Psi = \forall \pi . \Phi$, then

$$\mathsf{Value}(\Psi, H, \Pi, T, t) = \begin{cases} \mathsf{T} & \text{if for all } \mu \in H: \ \mathsf{Value}(\Phi, H, \Pi[\pi \mapsto \mu], T, t) \in \{\mathsf{T}, \mathsf{U}\}, \\ \mathsf{F} & \text{otherwise.} \end{cases}$$

Note that if a formula is closed, then its value is always either T or F.

For a hybrid system H, a trace assignment Π, a register valuation T, $t \in \mathbb{R}_{\geq 0}$, and a HyperSTL* formula Φ we can define the satisfaction relation $\models$ where

$$(H, \Pi, T, t) \models \Phi \text{ if and only if } \mathsf{Value}(\Phi, H, \Pi, T, t) = \mathsf{T}.$$

We say that a hybrid system H *satisfies a closed formula* Φ, denoted $H \models \Phi$, if and only if it holds that $(H, \Pi_\varnothing, T_0, 0) \models \Phi$, where $\Pi_\varnothing$ is the empty trace assignment and T_0 is the register valuation in which 0 is stored at every index.

Example 6.1. Let $H = \{\mu_1, \mu_2\}$ be a hybrid system that consists of two generalized timed traces, μ_1 with $\mathit{dom}(\mu_1) = \{0, 2, 4\}$ and μ_2 with $\mathit{dom}(\mu_2) = \{0, 1, 3\}$, where $\mu_1(t) = 0$ for all $t \in \{0, 2, 4\}$, and $\mu_2(0) = \mu_2(1) = 1$ and $\mu_2(3) = 0$.

Consider the HyperSTL* formula $\Phi_1 = \forall \pi_1 . \forall \pi_2 . \ \square_{[0,4]}(x_{\pi_1} = x_{\pi_2})$ that states that for every pair of timed traces and every time point in the interval $[0, 4]$ the value of the two traces must be equal (i.e., they agree on the value of variable x). We have that $H \not\models \Phi_1$ since the two traces differ at time point $t = 0$. If, on the other hand we consider the formula $\Phi_2 = \forall \pi_1 . \forall \pi_2 . \ \square_{[1,4]}(x_{\pi_1} = x_{\pi_2})$ obtained from Φ_1 by replacing the interval $[0, 4]$ by $[1, 4]$, we have that $H \models \Phi_2$. The justification behind this is that there is no time point in the interval $[1, 4]$ where we witness a violation of the atomic predicate $x_{\pi_1} = x_{\pi_2}$. In particular, in the time interval $[1, 4]$ there is no point at which both traces are defined.

Now, consider the HyperSTL* formula $\Phi_3 = \forall \pi_1 . \forall \pi_2 . \ \diamondsuit_{[0,4]}(x_{\pi_1} = x_{\pi_2})$ that states that for every pair of traces, in the interval $[0, 4]$ there *exists* a time point where the values of the two traces are the same. We have that $H \not\models \Phi_3$, as expected, since there is no point in this interval where both traces are defined and have the same value. Note that we also have $H \not\models \forall \pi_1 . \forall \pi_2 . \ \diamondsuit_{[1,4]}(x_{\pi_1} = x_{\pi_2})$ for the interval where the value of $x_{\pi_1} = x_{\pi_2}$ is undefined.

Finally, let $\Phi_4 = \forall \pi_1 . \forall \pi_2 . \ \diamondsuit_{[0,3]} *_1 \diamondsuit_{[0,1]} \, x_{\pi_1}^{*_1} = x_{\pi_2}$. The formula Φ_4 states that for every pair of traces there exists a time point t_1 in $[0, 3]$ such that there is a time point t_2 at

most 1 time unit later, such that the value of the first trace at t_1 is equal to the value of the second trace at time t_2. Here t_1 is the frozen time point per the semantics of the freeze operator $*$. We have that $H \models \Phi_4$. To see this, when π_1 is μ_1 and π_2 is μ_2 let $t_1 = 2$ and $t_2 = 3$, and when π_1 is μ_2 and π_2 is μ_1 let $t_1 = 3$ and $t_2 = 4$.

Remark 6.2. Due to the generality of our semantic domain, which generalizes both continuous signals and timed words, we have to address the issue of having to define the interpretation of HyperSTL* formulas over all time points in $\mathbb{R}_{\geq 0}$ while the considered traces might not be defined at all points. Furthermore, the semantics of the logic has to account for the fact that formulas, even atomic propositions, refer to different traces which are possibly defined over different time domains. To this end, we defined the function Value that assigns values in the set $\{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$. For instance, if $\mathsf{Value}(\alpha, H, \Pi, T, t) = \mathsf{U}$, then $\mathsf{Value}(\alpha \vee \neg\alpha, H, \Pi, T, t) = \mathsf{U}$. For the temporal operators, our semantics is reminiscent of that in [31], in the sense that for evaluating $\varphi_1 \,\mathcal{U}\, \varphi_2$ the subformula φ_1 is evaluated only in time points where its value is defined, and the time point where the obligation φ_2 must hold is one where its value is defined. The treatment in $\mathcal{S}$ is analogous. Our semantics interprets trace quantifiers over the traces for which the formula has a defined value.

Other temporal logics for timed hyperproperties face similar issues, which we discuss in Remark 6.4. The logic HyperSTL [43], on the other hand is not affected by such difficulties, since its semantics is defined over continuous signals defined over a whole interval.

Remark 6.3. In our definition of the semantics of $\varphi_1 \,\mathcal{U}_J\, \varphi_2$, similarly to [21] and [31], we account for the fact that in a dense time domain there might not exist a *first* time point where φ_2 is satisfied. Therefore we allow for φ_1 to be violated at intermediate time points as long as at those points the value of φ_2 is T and the constraint imposed by J is satisfied. More precisely, $\varphi_1 \,\mathcal{U}_J\, \varphi_2$ is T at time point t if there exists a time point $t' \geq t$ such that $t' - t \in J$, φ_2 is T at t', and for all intermediate points $t'' \in [t, t')$ it holds that if φ_1 is F at t'', then t'' must be such that $t'' - t \in J$ and φ_2 is T at t''. The analogous holds for $\mathcal{S}$.

Remark 6.4. Existing temporal logics for timed hyperproperties have also faced the challenge of dealing with timed traces that are defined over different sets of time points.

In [38] this leads to the consideration of two different semantics of their logic HyperMTL: an asynchronous semantics that does not require the time stamps in two timed traces to match, and a synchronous semantics in which the range of quantifiers is restricted to the traces that synchronize with the current trace assignment. The logic HyperMTL includes for each trace variable a Boolean constant $\top$ (true) indexed with that variable, which allows for expressing syntactically in formulas the requirement that the current time point is in the domain of the corresponding trace. In contrast, in our logic HyperSTL* we account for undefined values on the semantic level in the definition of the value function, and values at different points in time on different traces can be related via the freeze operator.

The authors of [12] provide an alternative logic HyperMTL by extending the logic MTL with quantifiers over traces in the point-wise semantics. The semantics of their logic has both a synchronous and an asynchronous layer. At the synchronous layer, traces are compared at the same points in time, and if a trace is undefined at a given point, the value at the closest previous event is used. At the asynchronous layer, an asynchronous version of the $\mathcal{U}$ operator allows for a bounded difference in the time points when the obligation of the *Until* formula is fulfilled in different traces. Our logic, on the other hand, allows for a general and flexible way of relating time points on different traces via the freeze operator.

6.3. **Expressing robust and hybrid cleanness.** Using the logic HyperSTL* we can express trace and hybrid conformance and robust and hybrid cleanness. We begin by first formalizing the conformance notions, and then provide the characterization of cleanness.

Let π_1 and π_2 be two trace variables, and let τ and ϵ be non-negative rational constants. We can express hybrid conformance with thresholds τ and ϵ, i.e., $\mathsf{HybridConf}_{\tau,\epsilon}$, as follows:

$$\varphi^{\mathsf{HybridConf}}_{\tau,\epsilon} \;=\; \Big(\square *_1\big(\diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon \vee \diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon\big)\Big) \wedge \Big(\square *_2\big(\diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_1}, X^{*_2}_{\pi_2}) \le \epsilon \vee \diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_1}, X^{*_2}_{\pi_2}) \le \epsilon\big)\Big)$$

where $d_{\mathcal{Y}}(X, X')$ is an arithmetic expression characterizing the metric $d_{\mathcal{Y}}$. Note that in the above formula the trace variables π_1 and π_2 are not quantified, and hence it is not closed.

Intuitively, the formula states that for every time point on the trace described by π_1 it holds that within τ time units in the past or in the future, there exists a point on the trace described by π_2 where the value is ϵ-close to the value of π_1 at the current time point, and symmetrically for the other direction with traces π_1 and π_2 swapped.

Proposition 6.5. *Let H be a deterministic hybrid system defined over a set of real-valued variables X such that $0 \in dom(\mu)$ for each $\mu \in H$. Let $\tau, \epsilon \ge 0$ be rational constants, and $\mu_1, \mu_2 \in H$. Let π_1 and π_2 be trace variables and $\Pi = \{\pi_1 \mapsto \mu_1, \pi_2 \mapsto \mu_2\}$. Then*

$$\mathsf{HybridConf}_{\tau,\epsilon}(\mu_1, \mu_2) \text{ if and only if } (H, \Pi, T_0, 0) \models \varphi^{\mathsf{HybridConf}}_{\tau,\epsilon}.$$

Proof. ($\Longrightarrow$) First, suppose that $\mathsf{HybridConf}_{\tau,\epsilon}(\mu_1, \mu_2)$.

By Definition 3.3, we have that for all $t_1 \in dom(\mu_1)$ there exists $t_2 \in dom(\mu_2)$ such that $|t_1 - t_2| \le \tau$ and $d_{\mathcal{Y}}(\mu_2(t_2), \mu_1(t_1)) \le \epsilon$. Hence, when $t_1 \in dom(\mu_1)$, we have that $\mathsf{Value}\big(*_1\,\big(\diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon \vee \diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon\big), H, \Pi, T_0, t_1\big) = \mathsf{T}$.

If $t_1 \notin dom(\mu_1)$, then, from the definition of the semantics of the operator $*_1$ we have that $\mathsf{Value}\big(*_1\,\big(\diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon \vee \diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon\big), H, \Pi, T_0, t_1\big) = \mathsf{U}$.

Thus, $\mathsf{Value}\big(\square *_1\big(\diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon \vee \diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon\big), H, \Pi, T_0, 0\big) = \mathsf{T}$.

$\mathsf{Value}\big(\square *_2\big(\diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_1}, X^{*_2}_{\pi_2}) \le \epsilon \vee \diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_1}, X^{*_2}_{\pi_2}) \le \epsilon\big), H, \Pi, T_0, 0\big) = \mathsf{T}$ can be shown by applying the same reasoning as above, this time for μ_2.

From the two facts we showed, we can conclude that $(H, \Pi, T_0, 0) \models \varphi^{\mathsf{HybridConf}}_{\tau,\epsilon}$.

($\Longleftarrow$) For the other direction, assume that $(H, \Pi, T_0, 0) \models \varphi^{\mathsf{HybridConf}}_{\tau,\epsilon}$.

Let $t_1 \in dom(\mu_1)$. Since $(H, \Pi, T_0, 0) \models \varphi^{\mathsf{HybridConf}}_{\tau,\epsilon}$, by the semantics of $\square$ we have that $\mathsf{Value}\big(*_1\,\big(\diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon \vee \diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon\big), H, \Pi, T_0, t_1\big) \in \{\mathsf{T}, \mathsf{U}\}$. Taking into account that we are considering the case when $t_1 \in dom(\mu_1)$, and that by definition

- $\mathsf{Value}\big(\diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon, H, \Pi, T_0, t_1\big) \neq U$ and
- $\mathsf{Value}\big(\diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon, H, \Pi, T_0, t_1\big) \neq U$,

we conclude that $\mathsf{Value}\big(*_1\big(\diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon \vee \diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon\big), H, \Pi, T_0, t_1\big) = \mathsf{T}$. Therefore, there exists $t_2 \in dom(\mu_2)$ such that $|t_1 - t_2| \le \tau$ and $d_{\mathcal{Y}}(\mu_2(t_2), \mu_1(t_1)) \le \epsilon$.

The reasoning for the other conjunct when $t_2 \in dom(\mu_2)$ is symmetric. We hence obtain $\mathsf{HybridConf}_{\tau,\epsilon}(\mu_1, \mu_2)$, which concludes the proof. □

As a special case, we can express $\mathsf{TraceConf}_{\epsilon}$ as

$$\varphi^{\mathsf{TraceConf}}_{\epsilon} \;=\; \Big(\square *_1\big(\diamondsuit_{[0,0]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \le \epsilon\big)\Big) \wedge \Big(\square *_2\big(\diamondsuit_{[0,0]}\, d_{\mathcal{Y}}(X_{\pi_1}, X^{*_2}_{\pi_2}) \le \epsilon\big)\Big).$$

Note that the formula $\varphi_\epsilon = \square\, d_{\mathcal{Y}}(X_{\pi_1}, X_{\pi_2}) \le \epsilon$ *does not* characterize trace conformance as it does not assert the requirement that the time domains of the two traces must be the

same. The formula $\varphi_\epsilon^{\mathsf{TraceConf}}$, on the other hand, requires that each time point where one of the traces is defined, must be matched by a value of the other trace at the same time point.

Proposition 6.6. *Let H be a deterministic hybrid system defined over a set of real-valued variables X such that $0 \in dom(\mu)$ for each $\mu \in H$. Let $\epsilon \geq 0$ be a rational constant, and $\mu_1, \mu_2 \in H$. Let π_1 and π_2 be trace variables and $\Pi = \{\pi_1 \mapsto \mu_1, \pi_2 \mapsto \mu_2\}$. Then*

$$\mathit{TraceConf}_\epsilon(\mu_1, \mu_2) \text{ if and only if } (H, \Pi, T_0, 0) \models \varphi_\epsilon^{\mathit{TraceConf}}.$$

Proof. ($\Longrightarrow$) First, suppose that $\mathsf{TraceConf}_\epsilon(\mu_1, \mu_2)$.

By Definition 3.3, we have that $dom(\mu_1) = dom(\mu_2)$ and for all $t_1 \in dom(\mu_1)$ it holds that $d_{\mathcal{Y}}(\mu_2(t_1), \mu_1(t_1)) \leq \epsilon$. This is equivalent to the conjunction of the following statements:

- for all $t_1 \in dom(\mu_1)$, it holds that $t_1 \in dom(\mu_2)$ and $d_{\mathcal{Y}}(\mu_2(t_1), \mu_1(t_1)) \leq \epsilon$, and
- for all $t_2 \in dom(\mu_2)$, it holds that $t_2 \in dom(\mu_1)$ and $d_{\mathcal{Y}}(\mu_1(t_2), \mu_2(t_2)) \leq \epsilon$.

Therefore, if $t \in dom(\mu_1) = dom(\mu_2)$ it holds that

$$\mathsf{Value}\big(*_1 \big(\Diamond_{[0,0]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \leq \epsilon \big), H, \Pi, T_0, t\big) = \mathsf{T} \text{ and}$$
$$\mathsf{Value}\big(*_2 \big(\Diamond_{[0,0]}\, d_{\mathcal{Y}}(X_{\pi_1}, X^{*_2}_{\pi_2}) \leq \epsilon \big), H, \Pi, T_0, t\big) = \mathsf{T}.$$

When $t \notin dom(\mu_1) = dom(\mu_2)$ we have that that

$$\mathsf{Value}\big(*_1 \big(\Diamond_{[0,0]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \leq \epsilon \big), H, \Pi, T_0, t\big) = \mathsf{U} \text{ and}$$
$$\mathsf{Value}\big(*_2 \big(\Diamond_{[0,0]}\, d_{\mathcal{Y}}(X_{\pi_1}, X^{*_2}_{\pi_2}) \leq \epsilon \big), H, \Pi, T_0, t\big) = \mathsf{U}.$$

Thus, $\mathsf{Value}\big(\varphi_\epsilon^{\mathsf{TraceConf}}, H, \Pi, T_0, 0\big) = \mathsf{T}$, which is what we had to prove.

($\Longleftarrow$) For the other direction, assume that $(H, \Pi, T_0, 0) \models \varphi_\epsilon^{\mathsf{TraceConf}}$.

Let $t_1 \in dom(\mu_1)$. We have that $\mathsf{Value}\big(*_1 \big(\Diamond_{[0,0]}\, d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \leq \epsilon \big), H, \Pi, T_0, t_1\big) = \mathsf{T}$. This implies that $t_1 \in dom(\mu_2)$ and $\mathsf{Value}\big(d_{\mathcal{Y}}(X_{\pi_2}, X^{*_1}_{\pi_1}) \leq \epsilon, H, \Pi, \{1 \mapsto t_1\}, t_1\big) = \mathsf{T}$.

Thus, we can conclude that for all $t_1 \in dom(\mu_1)$, it holds that $t_1 \in dom(\mu_2)$ and $d_{\mathcal{Y}}(\mu_2(t_1), \mu_1(t_1)) \leq \epsilon$. Analogously, we can show that for all $t_2 \in dom(\mu_2)$, it holds that $t_2 \in dom(\mu_1)$ and $d_{\mathcal{Y}}(\mu_1(t_2), \mu_2(t_2)) \leq \epsilon$. Hence, by Definition 3.3, $\mathsf{TraceConf}_\epsilon(\mu_1, \mu_2)$. □

We use the idea of the above encoding to define a closed $\mathsf{HyperSTL}^\star$ formula that characterizes hybrid cleanness, $\mathsf{HybridClean}(\tau_I, \epsilon_I, \tau_O, \epsilon_O)$, for deterministic hybrid systems.

For the rest of the section we consider hybrid systems H such that for every $\mu \in H$ it holds that $0 \in dom(\mu)$, that is, we assume that all traces are defined at time point 0.

Furthermore, we assume that the set of variables X defining the states of the hybrid system H contains an *explicit clock variable* c representing the current time, that is never reset. That is, c simply captures the time-stamps of the values of the GTTs in H. Formally, for every GTT $\mu \in H$, and every $t \in \mathbb{R}_{\geq 0}$, it holds that $\mu(t)(c) = t$. With that, the atomic propositions in AP can refer to the current time point, and the freeze operator captures the current time-stamp together with the current values of the other variables. Let $\alpha \in AP$ be an atomic proposition and $r, r_i \in \mathbb{R}_{\geq 0}$ for $i \in \mathcal{I}$ be non-negative real constants. We denote by $\alpha[r, r_1, \ldots, r_{|\mathcal{I}|}]$ the atomic predicate obtained from α by replacing each variable c_π by r, and each variable c^{*i}_π by r_i, for all $\pi \in V_{trace}$ and $i \in \mathcal{I}$. By the definition of the clock variable c, for every trace assignment Π, register valuation T, and $t \in \mathbb{R}_{\geq 0}$ we have that

$$\mathsf{Value}(\alpha, H, \Pi, T, t) = \mathsf{Value}(\alpha[t, T(1), \ldots, T(|\mathcal{I}|)], H, \pi, T, t),$$

when for every $\pi \in V_{trace}$ for which c_π occurs in α it holds that $t \in dom(\Pi(\pi))$ and for every $\pi \in V_{trace}$ and $i \in \mathcal{I}$ for which c^{*i}_π occurs in α it holds that $T(i) \in dom(\Pi(\pi))$.

Let τ_I and ϵ_I be non-negative rational constants defining the threshold values for the input conformance relation, and τ_O and ϵ_O be the ones for the output conformance.

Let π and π' be trace variables and $i, s, e \in \mathcal{I}$. First, we define the formulas

$$\begin{aligned}
\varphi^{\mathsf{matchI}}_{\tau_I,\epsilon_I}(\pi, i, \pi', s, e) &= *_i\big(\diamondsuit_{[0,\tau_I]}(d_I(X_{\pi'}, X^{*_i}_{\pi}) \le \epsilon_I \wedge c_{\pi'} \ge c^{*_s}_{\pi'} \wedge c_{\pi'} \le c^{*_e}_{\pi'}) \vee \\
&\qquad \diamondsuit_{[0,\tau_I]}(d_I(X_{\pi'}, X^{*_i}_{\pi}) \le \epsilon_I \wedge c_{\pi'} \ge c^{*_s}_{\pi'} \wedge c_{\pi'} \le c^{*_e}_{\pi'})\big), \\
\varphi^{\mathsf{matchO}}_{\tau_O,\epsilon_O}(\pi, i, \pi', s, e) &= *_i\big(\diamondsuit_{[0,\tau_O]}(d_O(X_{\pi'}, X^{*_i}_{\pi}) \le \epsilon_O \wedge c_{\pi'} \ge c^{*_s}_{\pi'} \wedge c_{\pi'} \le c^{*_e}_{\pi'}) \vee \\
&\qquad \diamondsuit_{[0,\tau_O]}(d_O(X_{\pi'}, X^{*_i}_{\pi}) \le \epsilon_O \wedge c_{\pi'} \ge c^{*_s}_{\pi'} \wedge c_{\pi'} \le c^{*_e}_{\pi'})\big),
\end{aligned}$$

where $d_I(X, X')$ and $d_O(X, X')$ are the arithmetic expressions characterizing the metrics on the sets of input and output values of the considered hybrid system.

Intuitively, the formula $\varphi^{\mathsf{matchI}}_{\tau_I,\epsilon_I}(\pi, i, \pi', s, e)$ evaluated at time point t and register valuation T is true if and only if there exists a time point $t' \in [t - \tau_I, t + \tau_I] \cap [T(s), T(e)]$ such that the input value at time t on the trace represented by π and the input value at time t' on the trace represented by π' are ϵ_I-close. The formula $\varphi^{\mathsf{matchO}}_{\tau_O,\epsilon_O}(\pi, i, \pi', s, e)$ states the same for the output values. The need to constrain the time t' where the match of the values at t must be found comes from the fact that in the definition of cleanness in Section 4, prefixes are compared using the predicate $\mathsf{PrefConf}$. Recall that $\mathsf{PrefConf}$ compares two prefixes μ_1 and μ_2 by requiring that there exist segments $\mu_1[t^s_1 \dots t^e_1]$ and $\mu_2[t^s_2 \dots t^e_2]$ obtained from them, that are conformant. In the above formulas, the frozen values of the clock variable c represent the end points of the interval for the trace assigned to π'.

Using the formula $\varphi^{\mathsf{matchI}}_{\tau_I,\epsilon_I}(\pi, i, \pi', s, e)$ we define the formula $\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1, \pi_2)$ which is true if and only if the current time point defines a pair of prefixes of the traces represented by π_1 and π_2 for which there exist hybrid conforming segments obtained from the prefixes by possibly removing a prefix/suffix of length at most τ_I or adding a suffix of length at most τ_I.

The formula is defined as a disjunction over the possible ways in which the end-points of the two segments are ordered on the time line. Let P be a set of 4-tuples of the indices $\{3, 4, 5, 6\} \subseteq \mathcal{I}$ such that $(s_1, e_1, s_2, e_2) \in P$ if and only if s_1, e_1, s_2, e_2 are pairwise different, and $s_1, s_2 \in \{3, 4\}$ and $e_1, e_2 \in \{5, 6\}$. For $\overline{p} = (s_1, e_1, s_2, e_2) \in P$, abusing notation we define $\overline{p}(s_1) = \overline{p}(e_1) = \pi_1$ and $\overline{p}(s_2) = \overline{p}(e_2) = \pi_2$. That is, the function $\overline{p}$ maps each endpoint index to the trace variable with which it is associated. We now define

$$\begin{aligned}
\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1, \pi_2) &= \bigvee_{\overline{p}=(s_1,e_1,s_2,e_2)\in P} \big(*_7\, \varphi_{\pi_1} \wedge *_7 \varphi_{\pi_2}\big) \\
\varphi_{\pi_1} &= \diamondsuit\big(c_{\overline{p}(3)} \le \tau_I \wedge *_3 \diamondsuit\big(c_{\overline{p}(4)} \le \tau_I \wedge \\
&\qquad *_4 \diamondsuit\big(c_{\overline{p}(5)} \ge (c^{*_7}_{\pi_1} - \tau_I) \wedge c_{\overline{p}(5)} \le (c^{*_7}_{\pi_1} + \tau_I) \wedge \\
&\qquad *_5 \diamondsuit\big(c_{\overline{p}(6)} \ge (c^{*_7}_{\pi_1} - \tau_I) \wedge c_{\overline{p}(6)} \le (c^{*_7}_{\pi_1} + \tau_I) \wedge *_6\, \varphi^{\mathsf{ConfI}}\big)\big)\big)\big) \\
\varphi_{\pi_2} &= \diamondsuit\big(c_{\overline{p}(3)} \le \tau_I \wedge *_3 \diamondsuit\big(c_{\overline{p}(4)} \le \tau_I \wedge \\
&\qquad *_4 \diamondsuit\big(c_{\overline{p}(5)} \ge (c^{*_7}_{\pi_2} - \tau_I) \wedge c_{\overline{p}(5)} \le (c^{*_7}_{\pi_2} + \tau_I) \wedge \\
&\qquad *_5 \diamondsuit\big(c_{\overline{p}(6)} \ge (c^{*_7}_{\pi_2} - \tau_I) \wedge c_{\overline{p}(6)} \le (c^{*_7}_{\pi_2} + \tau_I) \wedge *_6\, \varphi^{\mathsf{ConfI}}\big)\big)\big)\big) \\
\varphi^{\mathsf{ConfI}} &= \boxminus\Big(\big(c_{\pi_1} \ge c^{*_{s_1}}_{\pi_1} \wedge c_{\pi_1} \le c^{*_{e_1}}_{\pi_1}\big) \to \varphi^{\mathsf{matchI}}_{\tau_I,\epsilon_I}(\pi_1, 1, \pi_2, s_2, e_2)\Big) \wedge \\
&\qquad \boxminus\Big(\big(c_{\pi_2} \ge c^{*_{s_2}}_{\pi_2} \wedge c_{\pi_2} \le c^{*_{e_2}}_{\pi_2}\big) \to \varphi^{\mathsf{matchI}}_{\tau_I,\epsilon_I}(\pi_2, 2, \pi_1, s_1, e_1)\Big).
\end{aligned}$$

The conjunct $*_7\varphi_{\pi_1}$ handles the case when the current time point t (i.e., the last time point for the considered prefixes) is in the domain of π_1. The second conjunct handles the case when t is in the domain of π_2. If neither is the case, then the value of the conjunction is U. Since the formulas $*_7\varphi_{\pi_1}$ and $*_7\varphi_{\pi_2}$ differ only in the trace on which the time-point t is frozen, if both of them have a defined value, than these values are necessarily the same.

Each of φ_{π_1} and φ_{π_2} asserts the existence of a sequence of time points defining the compared segments of the two traces and their input conformance (formula φ^{ConfI}). The formula φ^{ConfI} captures the requirement that the two input prefixes ending at the current time point have segments (defined by the pairs of time points $c^{*s_1}_{\pi_1}$ and $c^{*e_1}_{\pi_1}$, and $c^{*s_2}_{\pi_2}$ and $c^{*e_2}_{\pi_2}$, respectively) that are hybrid conformant, as in Definition 4.6.

The formula $\varphi^{\mathsf{PrefConfO}}_{\tau_O,\epsilon_O}(\pi_1,\pi_2)$ is defined analogously using $\varphi^{\mathsf{matchO}}_{\tau_O,\epsilon_O}(\pi,i,\pi',s,e)$.

Proposition 6.7. *Let H be a deterministic hybrid system defined over a set of real-valued variables X that includes an explicit clock variable c, and such that $0 \in dom(\mu)$ for each $\mu \in H$. Let $\tau_I, \tau_O, \epsilon_I, \epsilon_O \geq 0$ be rational constants, and the predicates $\mathsf{PrefConf}_I$ and $\mathsf{PrefConf}_O$ be instantiated using $\mathit{HybridConf}_{\tau_I,\epsilon_I}$ and $\mathit{HybridConf}_{\tau_O,\epsilon_O}$ respectively. That is, let $\mathsf{Conf}_I = \mathit{HybridConf}_{\tau_I,\epsilon_I}$ and $\mathsf{Conf}_O = \mathit{HybridConf}_{\tau_O,\epsilon_O}$. Let $\mu_1, \mu_2 \in H$, let π_1 and π_2 be trace variables and $\Pi = \{\pi_1 \mapsto \mu_1, \pi_2 \mapsto \mu_2\}$ a trace assignment.*

(1) *If $t \in dom(\mu_1) \cup dom(\mu_2)$, then*
$\mathsf{PrefConf}_I(\mu_1^I, \mu_2^I, t)$ *is true if and only if* $\mathsf{Value}\big(\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2), H, \Pi, T_0, t\big) \in \{\mathsf{T}, \mathsf{U}\}$.

(2) *If $t \in dom(\mu_1) \cup dom(\mu_2)$, then*
$\mathsf{PrefConf}_O(\mu_1^O, \mu_2^O, t)$ *is true if and only if* $\mathsf{Value}\big(\varphi^{\mathsf{PrefConfO}}_{\tau_O,\epsilon_O}(\pi_1,\pi_2), H, \Pi, T_0, t\big) \in \{\mathsf{T}, \mathsf{U}\}$.

(3) *If $t \notin dom(\mu_1) \cup dom(\mu_2)$ then* $\mathsf{Value}\big(\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2), H, \Pi, T_0, t\big) = \mathsf{U}$.

(4) *If $t \notin dom(\mu_1) \cup dom(\mu_2)$ then* $\mathsf{Value}\big(\varphi^{\mathsf{PrefConfO}}_{\tau_O,\epsilon_O}(\pi_1,\pi_2), H, \Pi, T_0, t\big) = \mathsf{U}$.

Proof. We show (1), the proof for (2) is analogous.

($\Longrightarrow$) Suppose that $\mathsf{PrefConf}_I(\mu_1^I, \mu_2, ^I t)$ is true. By Definition 4.6, there exist $t_1^s, t_2^s \in [0, \tau_I]$ and $t_1^e, t_2^e \in [t-\tau_I, t+\tau_I]$ such that $\mathsf{Conf}_I(\mu_1^I[t_1^s \dots t_1^e], \mu_2^I[t_2^s \dots t_2^e])$ is true.

Let t_3, t_4, t_5, t_6 be a permutation of $t_1^s, t_2^s, t_1^e, t_2^e$ such that $t_3 \leq t_4 \leq t_5 \leq t_6$. Then, $t_3, t_4 \in [0, \tau_I]$ and $t_5, t_6 \in [t-\tau_I, t+\tau_I]$. Let $\overline{p} = (s_1, e_1, s_2, e_2) \in P$ be defined according to the permutation t_3, t_4, t_5, t_6, that is, $s_1 = i$ where t_1^s is t_i, and so on. Let $t_7 = t$. We define the register valuation $T = \{i \mapsto t_i \mid i \in \{3,4,5,6,7\}\}$.

Since $\mathsf{Conf}_I(\mu_1^I[t_1^s \dots t_1^e], \mu_2^I[t_2^s \dots t_2^e])$, we have that $\mathsf{Value}\big(\varphi^{\mathsf{ConfI}}, H, \Pi, T, t_6\big) = \mathsf{T}$.

Since $t \in dom(\mu_1) \cup dom(\mu_2)$, assume, without loss of generality that $t \in dom(\mu_1)$. By the choice of t_3, t_4, t_5, t_6 we have that $t_3 \leq t_4 \leq t_5 \leq t_6$, and $t_3, t_4 \in [0, \tau]$ and $t_5, t_6 \in [t-\tau_I, t+\tau_I]$ which, together with the definition of φ_{π_1} implies that $\mathsf{Value}\big(\varphi_{\pi_1}, H, \Pi, \{7 \mapsto t_7\}, t_7\big) = \mathsf{T}$.

If $t \in dom(\mu_2)$ we can show as above that the value of $*_7\varphi_{\pi_2}$ is T. Otherwise, the value of $*_7\varphi_{\pi_2}$ is U. Hence, we conclude that $\mathsf{Value}\big(\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2), H, \Pi, T_0, t\big) \in \{\mathsf{T}, \mathsf{U}\}$.

($\Longleftarrow$) Now, suppose that $\mathsf{Value}\big(\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2), H, \Pi, T_0, t\big) \in \{\mathsf{T}, \mathsf{U}\}$. Hence, there exists $\overline{p} = (s_1, e_1, s_2, e_2) \in P$ for which we have $\mathsf{Value}\big(\big(*_7\,\varphi_{\pi_1} \wedge *_7\varphi_{\pi_2}\big), H, \Pi, T_0, t\big) \in \{\mathsf{T}, \mathsf{U}\}$. Since $t \in dom(\mu_1) \cup dom(\mu_2)$, assume, without loss of generality, that $t \in dom(\mu_1)$. Then we must have that $\mathsf{Value}\big(*_7\ \varphi_{\pi_1}, H, \Pi, T_0, t\big) = \mathsf{T}$. Let $t_7 = t$. Then, $\mathsf{Value}\big(\varphi_{\pi_1}, H, \Pi, \{7 \mapsto t_7\}, t\big) = \mathsf{T}$. Therefore, according to the definition of φ_{π_1} we have that there exist $t_3 \leq t_4 \leq t_5 \leq t_6$ such that $t_3, t_4 \in [0, \tau]$ and $t_5, t_6 \in [t-\tau_I, t+\tau_I]$, and, furthermore, $\mathsf{Value}\big(\varphi^{\mathsf{ConfI}}, H, \Pi, T, t_6\big) = \mathsf{T}$ for the valuation $T = \{i \mapsto t_i \mid i \in \{3,4,5,6,7\}\}$.

We define $t_1^s, t_2^s, t_1^e, t_2^e$ to be the permutation of t_3, t_4, t_5, t_6 determined by $\overline{p}$, that is $t_1^s = t_{s_1}$, and so on. Then, since $\mathsf{Value}\big(\varphi_{\pi_1}, H, \Pi, \{7 \mapsto t_7\}, t\big) = \mathsf{T}$, we have that $t_1^s \leq t_1^e$, $t_2^s \leq t_2^e$, and $t_1^s, t_2^s \in [0, \tau_I]$ and $t_1^e, t_2^e \in [t-\tau_I, t+\tau_I]$. Hence, $\mathsf{Value}\big(\varphi^{\mathsf{ConfI}}, H, \Pi, T, t_6\big) = \mathsf{T}$ implies that $\mathsf{Conf}_I(\mu_1^I[t_1^s \dots t_1^e], \mu_2^I[t_2^s \dots t_2^e])$, which allows us to conclude that $\mathsf{PrefConf}_I(\mu_1^I, \mu_2^I, t)$.

(3) and (4) follow directly from the semantics of the operators $\wedge$ and $*$. □

We now define the formula characterizing hybrid cleanness as

$$\Phi^{\mathsf{HybridClean}}_{\tau_I,\epsilon_I,\tau_O,\epsilon_O} = \forall\pi_1.\forall\pi_2.\ \square\Big(\big(\boxminus\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2)\big) \rightarrow \varphi^{\mathsf{PrefConfO}}_{\tau_O,\epsilon_O}(\pi_1,\pi_2)\Big).$$

Proposition 6.8. *Let H be a deterministic hybrid system defined over a set of real-valued variables X that includes an explicit clock variable c, and such that $0 \in dom(\mu)$ for each $\mu \in H$. Let $\tau_I, \tau_O, \epsilon_I, \epsilon_O \geq 0$ be rational constants. It holds that*

$$H \text{ is } \mathsf{HybridClean}(\tau_I,\epsilon_I,\tau_O,\epsilon_O) \text{ if and only if } H \models \Phi^{\mathsf{HybridClean}}_{\tau_I,\epsilon_I,\tau_O,\epsilon_O}.$$

Proof. ($\Longrightarrow$) First, suppose that H is $\mathsf{HybridClean}(\tau_I,\epsilon_I,\tau_O,\epsilon_O)$. Let $\mu_1,\mu_2 \in H$ be two arbitrarily chosen traces. Let π_1 and π_2 be trace variables, and let $\Pi = \{\pi_1 \mapsto \mu_1, \pi_2 \mapsto \mu_2\}$.

Let $t \geq 0$ be an arbitrary time point. If $\mathsf{Value}\big(\varphi^{\mathsf{PrefConfO}}_{\tau_O,\epsilon_O}(\pi_1,\pi_2), H, \Pi, T_0, t\big) \in \{\mathsf{T},\mathsf{U}\}$, then it holds that $\mathsf{Value}\Big(\big(\boxminus\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2)\big) \rightarrow \varphi^{\mathsf{PrefConfO}}_{\tau_O,\epsilon_O}(\pi_1,\pi_2), H, \Pi, T_0, t\Big) \in \{\mathsf{T},\mathsf{U}\}$. If, on the other hand we have that $\mathsf{Value}\big(\varphi^{\mathsf{PrefConfO}}_{\tau_O,\epsilon_O}(\pi_1,\pi_2), H, \Pi, T_0, t\big) = \mathsf{F}$, then it holds that $t \in dom(\mu_1) \cup dom(\mu_2)$ and by Proposition 6.7 we have that $\mathsf{PrefConf}_O(\mu_1^O, \mu_2^O, t)$ is false. According to Definition 4.7, this means that there exists $t' \leq t$ such that $\mathsf{PrefConf}_I(\mu_1^I, \mu_2^I, t')$ is false. Applying Proposition 6.7 we obtain for the time point t' that $\mathsf{Value}\big(\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2), H, \Pi, T_0, t'\big) = \mathsf{F}$, and hence $\mathsf{Value}\big(\boxminus\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2), H, \Pi, T_0, t\big) = \mathsf{F}$. This implies that $\mathsf{Value}\Big(\big(\boxminus\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2)\big) \rightarrow \varphi^{\mathsf{PrefConfO}}_{\tau_O,\epsilon_O}(\pi_1,\pi_2), H, \Pi, T_0, t\Big) = \mathsf{T}$.

Therefore, since $t \geq 0$ was chosen arbitrarily, $(H, \Pi, T_0, 0) \models \square\big(\big(\boxminus\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2)\big) \rightarrow \varphi^{\mathsf{PrefConfO}}_{\tau_O,\epsilon_O}(\pi_1,\pi_2)\big)$. Since μ_1 and μ_2 were arbitrary, we conclude $(H, \Pi_\varnothing, T_0, 0) \models \Phi^{\mathsf{HybridClean}}_{\tau_I,\epsilon_I,\tau_O,\epsilon_O}$.

($\Longleftarrow$) Now, suppose that $(H, \Pi_\varnothing, T_0, 0) \models \Phi^{\mathsf{HybridClean}}_{\tau_I,\epsilon_I,\tau_O,\epsilon_O}$. Let $\mu_1,\mu_2 \in H$ be two arbitrarily chosen traces. Let π_1 and π_2 be trace variables, and define $\Pi = \{\pi_1 \mapsto \mu_1, \pi_2 \mapsto \mu_2\}$.

Let $t \in dom(\mu_1) \cup dom(\mu_2)$ be such that for every $t' \leq t$ with $t' \in dom(\mu_1) \cup dom(\mu_2)$ it holds that $\mathsf{PrefConf}_I(\mu_1^I, \mu_2^I, t')$ is true. By Proposition 6.7, for every such t' we have $\mathsf{Value}\big(\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2), H, \Pi, T_0, t'\big) \in \{\mathsf{T},\mathsf{U}\}$. If $t' \notin dom(\mu_1) \cup dom(\mu_2)$, we have $\mathsf{Value}\big(\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2), H, \Pi, T_0, t'\big) = \mathsf{U}$. Thus, $\mathsf{Value}(\boxminus\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2), \Pi, T_0, t) = \mathsf{T}$.

By assumption, $\mathsf{Value}\Big(\big(\boxminus\varphi^{\mathsf{PrefConfI}}_{\tau_I,\epsilon_I}(\pi_1,\pi_2)\big) \rightarrow \varphi^{\mathsf{PrefConfO}}_{\tau_O,\epsilon_O}(\pi_1,\pi_2), H, \Pi, T_0, t\Big) \in \{\mathsf{T},\mathsf{U}\}$. Therefore, $\mathsf{Value}\big(\varphi^{\mathsf{PrefConfO}}_{\tau_O,\epsilon_O}(\pi_1,\pi_2), H, \Pi, T_0, t\big) \in \{\mathsf{T},\mathsf{U}\}$. Since $t \in dom(\mu_1) \cup dom(\mu_2)$, by Proposition 6.7, this implies that $\mathsf{PrefConf}_O(\mu_1^O, \mu_2^O, t)$ is true. Since $t \in dom(\mu_1) \cup dom(\mu_2)$ was chosen arbitrarily, we conclude that $\mathsf{HybridClean}(\tau_I,\epsilon_I,\tau_O,\epsilon_O)$. □

In the special case when $\tau_I = \tau_O = 0$, i.e., when we consider robust cleanness, the characterization of cleanness is simpler, since in the definition of PrefConf we only need to consider the actual compared prefixes (and not their truncated or extended versions) because the timing deviation is 0. As per the definition of robust cleanness, we consider PrefConf instantiated with TraceConf.

Thus, we can characterize robust cleanness as

$$\Phi^{\mathsf{RobustClean}}_{\epsilon_I,\epsilon_O} = \forall\pi_1.\forall\pi_2.\ \square\Big(\big(\boxminus\varphi^I_{\epsilon_I}\big) \rightarrow \varphi^O_{\epsilon_O}\Big), \text{ where}$$

$$\begin{aligned}\varphi^I_{\epsilon_I} &= \big(*_1 \diamondsuit_{[0,0]}\, d_I(X_{\pi_2}, X^{*_1}_{\pi_1}) \leq \epsilon_I\big) \wedge \big(*_2 \diamondsuit_{[0,0]}\, d_I(X_{\pi_1}, X^{*_2}_{\pi_2}) \leq \epsilon_I\big) \text{ and}\\ \varphi^O_{\epsilon_O} &= \big(*_3 \diamondsuit_{[0,0]}\, d_O(X_{\pi_2}, X^{*_3}_{\pi_1}) \leq \epsilon_O\big) \wedge \big(*_4 \diamondsuit_{[0,0]}\, d_O(X_{\pi_1}, X^{*_4}_{\pi_2}) \leq \epsilon_O\big).\end{aligned}$$

Proposition 6.9. *Let H be a deterministic hybrid system defined over a set of real-valued variables X such that $0 \in dom(\mu)$ for each $\mu \in H$. Let $\epsilon_I, \epsilon_O \geq 0$ be rational constants. It holds that*

$$H \text{ is } \mathsf{RobustClean}(\epsilon_I, \epsilon_O) \text{ if and only if } H \models \Phi^{\mathsf{RobustClean}}_{\epsilon_I,\epsilon_O}.$$

Proof. ($\Longrightarrow$) First, suppose that H is $\mathsf{RobustClean}(\epsilon_I, \epsilon_O)$. Let $\mu_1, \mu_2 \in H$ be two arbitrarily chosen traces. Let π_1 and π_2 be trace variables, and let $\Pi = \{\pi_1 \mapsto \mu_1, \pi_2 \mapsto \mu_2\}$.

Let $t \geq 0$ be an arbitrary time point. If $\mathsf{Value}\big(\varphi^O_{\epsilon_O}, H, \Pi, T_0, t\big) \in \{\mathsf{T}, \mathsf{U}\}$, then it holds that $\mathsf{Value}\big(\square\big(\big(\boxminus\varphi^I_{\epsilon_I}\big) \rightarrow \varphi^O_{\epsilon_O}\big), H, \Pi, T_0, t\big) \in \{\mathsf{T}, \mathsf{U}\}$. If, on the other hand we have that $\mathsf{Value}\big(\varphi^O_{\epsilon_O}, H, \Pi, T_0, t\big) = \mathsf{F}$, then it holds that at least one of the conjuncts of $\varphi^O_{\epsilon_O}$ is false. Suppose that, without loss of generality, $\mathsf{Value}\big(\ *_3 \diamondsuit_{[0,0]}\, d_O(X_{\pi_2}, X^{*_3}_{\pi_1}) \leq \epsilon_O, H, \Pi, T_0, t\big) = \mathsf{F}$. Hence, $t \in dom(\mu_1) \cup dom(\mu_2)$ and according to Definition 4.7, this means that there exists $t' \leq t$ such that $\mathsf{Value}\big(\varphi^I_{\epsilon_I}, H, \Pi, T_0, t'\big) = \mathsf{F}$. Therefore, $\mathsf{Value}\big(\boxminus\varphi^I_{\epsilon_I}, H, \Pi, T_0, t\big) = \mathsf{F}$. Thus, in this case we obtain that $\mathsf{Value}\big(\big(\boxminus\varphi^I_{\epsilon_I}\big) \rightarrow \varphi^O_{\epsilon_O}, H, \Pi, T_0, t\big) = \mathsf{T}$.

Therefore, since $t \geq 0$ was chosen arbitrarily, we have $(H, \Pi, T_0, 0) \models \square\big(\big(\boxminus\varphi^I_{\epsilon_I}\big) \rightarrow \varphi^O_{\epsilon_O}\big)$. Since μ_1 and μ_2 were arbitrary, we can conclude $(H, \Pi_\varnothing, T_0, 0) \models \Phi^{\mathsf{RobustClean}}_{\tau_I,\epsilon_I,\tau_O,\epsilon_O}$.

($\Longleftarrow$) Now, suppose that $(H, \Pi_\varnothing, T_0, 0) \models \Phi^{\mathsf{RobustClean}}_{\tau_I,\epsilon_I,\tau_O,\epsilon_O}$. Let $\mu_1, \mu_2 \in H$ be two arbitrarily chosen traces. Let π_1 and π_2 be trace variables, and define $\Pi = \{\pi_1 \mapsto \mu_1, \pi_2 \mapsto \mu_2\}$.

Let $t \in dom(\mu_1) \cup dom(\mu_2)$ be such that for every $t' \leq t$ with $t' \in dom(\mu_1) \cup dom(\mu_2)$ it holds that $\mathsf{Value}\big(\varphi^I_{\epsilon_I}, H, \Pi, T_0, t'\big) = \mathsf{T}$. If $t' \notin dom(\mu_1) \cup dom(\mu_2)$, we have $\mathsf{Value}\big(\varphi^I_{\epsilon_I}, H, \Pi, T_0, t'\big) = \mathsf{U}$. Therefore, we have that $\mathsf{Value}(\boxminus\varphi^I_{\epsilon_I}, \Pi, T_0, t) = \mathsf{T}$.

By assumption, $\mathsf{Value}\big(\big(\boxminus\varphi^I_{\epsilon_I}\big) \rightarrow \varphi^O_{\epsilon_O}, H, \Pi, T_0, t\big) \in \{\mathsf{T}, \mathsf{U}\}$. Therefore, we obtain that $\mathsf{Value}\big(\varphi^O_{\epsilon_O}, H, \Pi, T_0, t\big) = \mathsf{T}$, since $t \in dom(\mu_1) \cup dom(\mu_2)$. Since t was chosen arbitrarily, we can conclude that $\mathsf{RobustClean}(\epsilon_I, \epsilon_O)$. □

6.4. **Monitoring $\mathsf{HyperSTL}^\star_{fin}$ over finite-length real-valued signals.** We now consider the fragment $\mathsf{HyperSTL}^\star_{fin}$ and describe a method for offline monitoring of $\mathsf{HyperSTL}^\star_{fin}$ properties on finite sets of finite-length signals.

If the given traces are finite timed words, we can obtain from them piecewise linear signals by linear interpolation, or piecewise constant signals by fixing the value for each half-closed interval between time points to be the value at the starting point of this interval.

If the signals in the set are of different time length, we take the minimum length across the set, and consider the traces up to that length. Thus, we ensure that for some B, all traces are defined over the interval $[0, B]$. Furthermore, we only consider formulas in $\mathsf{HyperSTL}^\star_{fin}$ for which the bounds of the temporal operators are such that every subformula has a defined value when the formula is evaluated over traces with time domain $[0, B]$.

Our method handles the trace quantifiers similarly to the algorithm for offline monitoring of HyperLTL formulas on finite traces given in [27, 29]. The method iterates over tuples of generalized timed traces. The arity of the tuples is determined by the quantifier prefix of the formula. For instance, for monitoring a formula of the form $\Phi = \forall\pi_1 \ldots \forall\pi_n \exists\pi'_1 \ldots \exists\pi'_m.\varphi$ we will evaluate φ on tuples of GTTs of arity $n + m$, to either determine that Φ is satisfied over the given set of traces, or return an n-tuple witnessing a violation. In order to evaluate the trace-quantifier-free formula φ on an $(n + m)$- tuple of traces we compute a satisfaction evidence by using the method proposed in [14]. Note that unlike [27, 29] we consider finite traces and formulas with bounded temporal operators. Since we assume that the length of the traces is sufficient for all subformulas of a given $\mathsf{HyperSTL}^\star_{fin}$ formula of interest to have a defined value, the truth value of the quantifier-free part of the formula is defined. As we

consider a fixed set of recorded traces, we can check offline the satisfaction of $\mathsf{HyperSTL}^{\star}_{fin}$ formulas with arbitrary quantifier alternations similarly to [27, 29], as outlined above.

Let φ be a trace-quantifier-free formula. Let H be a finite set of $\mathbb{R}^l$-valued finite-length signals, and let $K \in H^k$ be a tuple of traces of arity k. Then, we can interpret K as a real-valued signal κ of order $k \times l$. The *satisfaction set* of the formula φ over the signal κ is defined analogously to [14]. Similarly to [14], we assume that the traces in H are piecewise linear and that the atomic predicates are linear, in order to make the computation of the satisfaction set tractable. Note that the atomic predicates used in the characterization of hybrid cleanness in Section 6.3 are linear when the expressions $d_I(X, X')$ and $d_O(X, X')$ are linear. The explicit clock variable defines a linear signal. Clearly, if the traces in H are piecewise linear, then so is the signal κ. Thus, the satisfaction set for φ can be calculated effectively, represented as convex polytopes.

The satisfaction set for the formula φ given the signal κ is a subset of $\mathbb{R}_{\geq 0} \times \mathbb{R}^{\mathcal{I}}_{\geq 0}$, defined inductively with respect to structure of φ. Here, a signal κ of order $k \times l$ is interpreted as a trace assignment for k trace variables, in which each trace variable is assigned a GTT in the form of a real-valued signal of order l.

$$
\begin{array}{lcl}
\mathsf{Sat}(\alpha, \kappa) & = & \{(t,T) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^{\mathcal{I}}_{\geq 0} \mid \mathsf{Value}(\alpha, H, \kappa, T, t) = \mathsf{T}\}; \\
\mathsf{Sat}(\top, \kappa) & = & \mathbb{R}_{\geq 0} \times \mathbb{R}^{\mathcal{I}}_{\geq 0}; \\
\mathsf{Sat}(\neg\varphi', \kappa) & = & (\mathbb{R}_{\geq 0} \times \mathbb{R}^{\mathcal{I}}_{\geq 0}) \setminus \mathsf{Sat}(\varphi', \kappa); \\
\mathsf{Sat}(\varphi_1 \vee \varphi_2, \kappa) & = & \mathsf{Sat}(\varphi_1, \kappa) \cup \mathsf{Sat}(\varphi_2, \kappa); \\
\mathsf{Sat}(\varphi_1 \,\mathcal{U}_{[a,b]}\, \varphi_2, \kappa) & = & \{(t,T) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^{\mathcal{I}}_{\geq 0} \mid \exists t' \in [t+a, t+b] : (t', T) \in \mathsf{Sat}(\varphi_2, \kappa) \wedge \\
 & & \quad \forall t'' \in [t, t') : (t'', T) \in \mathsf{Sat}(\varphi_1, \kappa) \cup \mathsf{Sat}(\varphi_2, \kappa) \cap [t+a, t+b]\}; \\
\mathsf{Sat}(\varphi_1 \,\mathcal{S}_{[a,b]}\, \varphi_2, \kappa) & = & \{(t,T) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^{\mathcal{I}}_{\geq 0} \mid \exists t' \in [t-b, t-a] : (t', T) \in \mathsf{Sat}(\varphi_2, \kappa) \wedge \\
 & & \quad \forall t'' \in (t', t] : (t'', T) \in \mathsf{Sat}(\varphi_1, \kappa) \cup \mathsf{Sat}(\varphi_2, \kappa) \cap [t-b, t-a]\}; \\
\mathsf{Sat}(*_i\varphi', \kappa) & = & \{(t,T) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^{\mathcal{I}}_{\geq 0} \mid (t, T[i \mapsto t]) \in \mathsf{Sat}(\varphi', \kappa)\}.
\end{array}
$$

Once the satisfaction sets for the atomic predicates appearing in the given formula have been computed, the satisfaction sets for the composite formulas can be constructed by following the inductive definition above. Under the assumptions we made above about the signals and the atomic predicates, the satisfaction sets for the atomic predicates can be computed directly using the method described in [14]. For further details, we refer to [14].

In order to perform monitoring for the formula $\Phi^{\mathsf{HybridClean}}_{\tau_I, \epsilon_I, \tau_O, \epsilon_O}$ defined in Section 6.3 , we bound the temporal operators based on the signal length B, and consider the case when $\tau_I > 0$ and $\tau_O > 0$ which ensures that the intervals in the operators are non-singular.

7. Case Study

In this section we evaluate the proposed notion of conformance-based cleanness in a real application context, known as the Diesel Emissions Scandal [10, 37, 40, 9, 18, 17, 8]: Starting in fall 2015, millions of diesel cars were found being equipped with defeat devices reducing the effectiveness of emission cleaning systems during real-world usage — in contrast to the regulator-defined driving scenarios on a chassis dynamometer, where the amount of emitted pollutants stay well below the applicable limits.

It was soon suspected, that the singularities of the testing procedure were straightforward to identify and hence made cheating easy; in particular, there was only a single test cycle for testing emission cleaning systems, to be executed under very particular conditions. In the European Union, this was the *New European Driving Cycle* (NEDC) [48], the speed

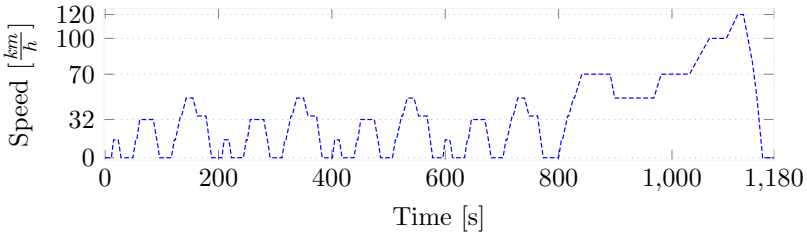

Figure 3: Left: New European Driving Cycle (NEDC); Right: Test Setup with Nissan NV200 Evalia on a chassis dynamometer attached to a PEMS.

profile of which is shown in Fig. 3. The NEDC consists of four repetitions of an elementary *urban driving cycle* (UDC) followed by one *extra urban driving cycle* (EUDC). Each test run is preceded by a preconditioning phase (PreCon), in which three EUDCs are driven consecutively. Between PreCon and the test, the vehicle has to cool down for 6 to 36 hours at an ambient temperature between 20 and 30 degrees Celsius.

Robust cleanness gives us a way of deriving additional test cycles that are reasonable w.r.t. the official NEDC. For a concrete context, "reasonable" is defined by an accompanying formally defined *contract*. The contributions of this paper enable us to go beyond previous work [10] where a contract allowed inputs and outputs to deviate only in the value domains, but not in the time domain.

7.1. **Experimental Setup.** Our empirical studies apply the theory developed in the previous sections in a very specific setting. The system under test is a Nissan NV200 Evalia equipped with a Renault 1.5 dci (110hp) diesel engine and approved w.r.t. regulation *Euro 6b*. All tests were conducted in November 2020. As shown in Fig. 3, the car is fixed on a chassis dynamometer and attached to a portable emissions measurement system (PEMS) in preparation for the test. The PEMS is connected to the onboard diagnostics (OBD) interface of the vehicle. During a test, the PEMS measures the amount of several gases at the end of the car's exhaust pipe and logs the data received from OBD. The PEMS is able to internally synchronise the times of gas measurements and OBD data. We will not consider the internal PEMS retiming and instead analyse the final data set. As input, we consider the OBD speed data, as output, the sum of emitted NO and NO_2 (abbreviated as NO_x). The input and output is sampled by 1 Hz. The amount of NO_x emitted along different runs is comparable only to a limited extend. This is because the emission cleaning system used can have internal regeneration phases, which – from an external observer perspective – are triggered nondeterministically. Hence, for the formal analysis we consider the accumulated output over 1180 seconds (the length of a complete NEDC). This is also the value decisive for type approval according to the official regulations.

Formally, a contract specifying *clean* behaviour for a system is given as a tuple. Previous work [10] proposes the concrete contract $\mathcal{C}_t = \langle d_I, d_O, \epsilon_I, \epsilon_O \rangle$ for diesel doping tests, where $d_I(i_1, i_2) \triangleq |i_1 - i_2|$ and $d_O(o_1, o_2) \triangleq |o_1 - o_2|$, and $\epsilon_I = 15$ km/h and $\epsilon_O = 180$ mg/km. $\mathcal{C}_t$ is based on robust cleanness, i.e., $\mathsf{TraceConf}_{\epsilon_I}$ and $\mathsf{TraceConf}_{\epsilon_O}$ conformance for inputs and outputs. In the following, we will add the designated conformance notions to the tuple, i.e., instead of $\mathcal{C}_t$ we write $\mathcal{C} = \langle d_I, d_O, \mathsf{TraceConf}_{\epsilon_I}, \mathsf{TraceConf}_{\epsilon_O} \rangle$ (implicitly encoding ϵ_I and ϵ_O). In $\mathcal{C}$ the degree of deviation is constrained by a threshold on the pointwise difference of the new and the reference test input. Comparisons of data at different time points are not

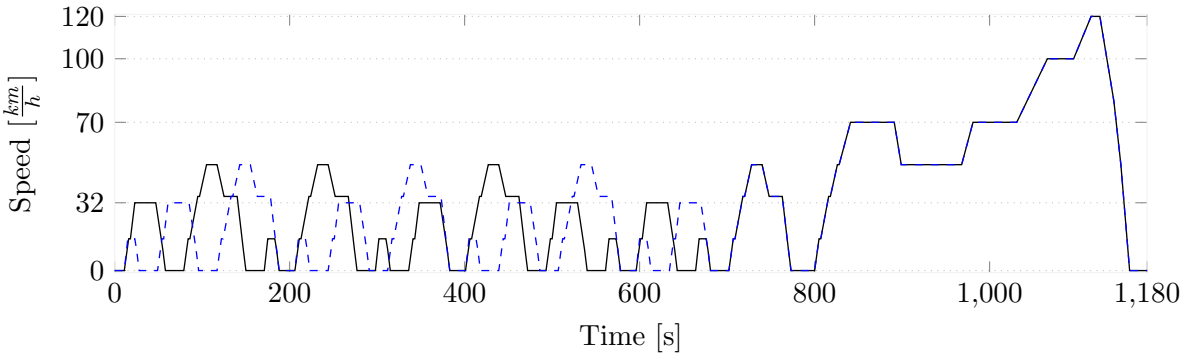

Figure 4: PermNEDC (solid, black line) compared to NEDC (dashed, blue line)

possible. The theory of conformance-based doping tests developed in this paper improves upon the previous testing methodology by enabling variation in the time domain. This gives us the possibility of reordering NEDC segments, lengthening a test beyond the time limits of the original NEDC, and of increased tolerance regarding human-caused input distortions during test execution. With the parameters $d_I, d_O, \epsilon_I, \epsilon_O$ and $\mathsf{TraceConf}_{\epsilon_O}$ in $\mathcal{C}$ fixed, we show how adaptations of $\mathcal{C}$ with different input conformance relations are of benefit for software doping analysis.

NEDC Permutations: Based on the conformance notions in this work, we propose a new test cycle PermNEDC in which NEDC segments are permuted on the time axis. Fig. 4 shows the test cycle. In each of the four UDC segments the three non-zero speed-phases are permuted. The transformation from NEDC to PermNEDC can be described by a retiming function r_p. An explicit definition of r_p is space consuming, hence we omit it. Along with the new cycle, we propose two suitable variants of contract $\mathcal{C}$ with different input conformances. Neither input conformance is constrained by a time threshold; in other words, $\tau = \infty$, so we omit τ in the index.

- Contract $\mathcal{C}_a$ is as $\mathcal{C}$, but entails input conformance $\mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_a}$, where $\mathsf{Ret}_a = \{(r, r^{-1}) \mid r \in \mathcal{T} \to \mathcal{T}$ and r is total and bijective$\}$ is the family of retimings that allows any reordering of the NEDC inputs. Notably, no inputs can be added or removed.
- Contract $\mathcal{C}_p$ adjusts $\mathcal{C}$ by enforcing input conformance $\mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_p}$, where $\mathsf{Ret}_p = \{(r_p, r_p^{-1})\}$ is the family of retimings that only allows the particular retiming r_p used to design the test cycle as discussed above. This input conformance is stricter than $\mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_a}$ above; it enforces that PermNEDC is not permuted any further by the driver.

NEDC Lengthening: Conformance-based doping tests can run longer than the NEDC; this is not possible with robust cleanness. We propose the test cycle DoubleNEDC, which consists of two consecutive NEDCs. In contrast to all other test cycles in this paper, DoubleNEDC produces two outputs: the first after 1180 seconds, and the second after 2360 seconds. The first half of this cycle is a classical "cold" NEDC (i.e., the engine cooled down before the test execution). The second half is a "hot" NEDC, since the cool-down phase was implicitly skipped. Also, the PreCon phase is skipped implicitly; there is only a single EUDC (instead of three) prior to the second NEDC. The inputs of both NEDCs can be compared by using the retiming functions id and $r_d = \lambda\, t.\ t \ \mathit{mod}\ 1180$. r_d maps time points of the global "test case clock" to the local time point in the NEDC time domain. DoubleNEDC requires to adapt contract $\mathcal{C}$ to $\mathcal{C}_d$ by replacing robust cleanness

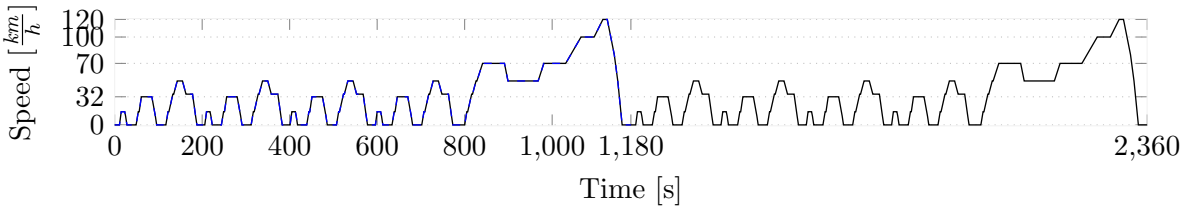

Figure 5: DoubleNEDC (solid, black line) compared to NEDC (dashed, blue line)

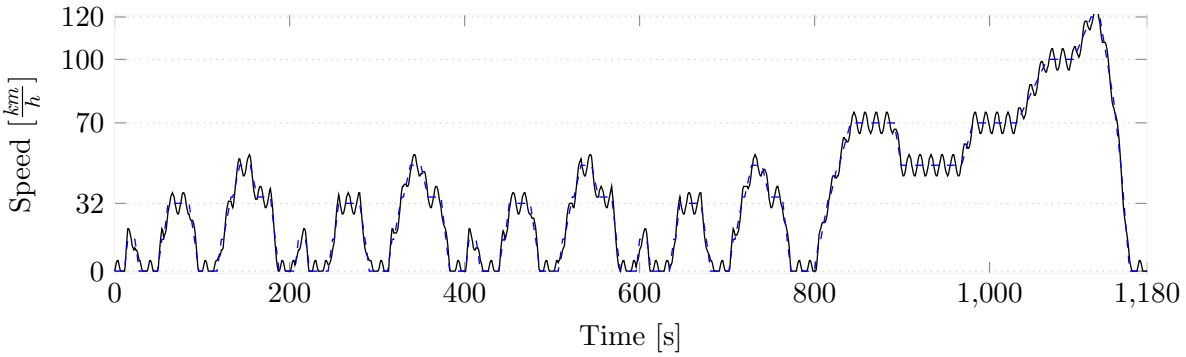

Figure 6: SineNEDC (solid, black line) compared to NEDC (dashed, blue line)

by cleanness with synchronised retiming: it entails input conformance $\mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_d}$, which allows only id and r_d as retiming functions, i.e., $\mathsf{Ret}_d = \{(\mathsf{id}, r_d)\}$. Similarly, $\mathcal{C}_d$ must include the synchronisation retiming function $\mathsf{Sync}_d(r_1, r_2) = (\mathsf{id}, r_d)$, which enforces that both DoubleNEDC outputs are compared to the single NEDC output (independent of the input retimings r_1 and r_2).

Human Time Imprecision Tolerance: Diesel doping tests are executed by humans driving a car. Humans tend to make mistakes when driving. Mistakes can be the over- or undershooting of the targeted speed (the error is on the value axis), or accelerations or decelerations happening too early or too late (the error is a shift on the time axis), or superpositions thereof. To compensate for both value and time errors, we use hybrid conformance. As a formal contract, this would be expressed by a variant of $\mathcal{C}$, in which the input conformance is replaced by $\mathsf{HybridConf}_{\tau_I,\epsilon_I}$ for some $\tau_I > 0$. For the purpose of demonstration, we will later analyse several such variants of $\mathcal{C}$, each variant with a unique value for τ_I and ϵ_I, i.e., we consider the contract $\mathcal{C}(\tau_I, \epsilon_I)$ parametrised in τ_I and ϵ_I. Concrete values for τ_I and ϵ_I must be specified when using the contract.

A test cycle that reflects drivings rich of acceleration and deceleration phases—and is hence particularly prone to human driving errors—is SineNEDC [10]. SineNEDC is defined as the NEDC superimposed by a sine curve, formally $\text{SineNEDC}(t) = \max\{0, \text{NEDC}(t) + 5\sin(0.5t)\}$, with a maximum input deviation from NEDC of 5km/h, compare Fig. 6. We will evaluate SineNEDC under several variants of $\mathcal{C}(\tau_I, \epsilon_I)$.

Human time imprecision is as yet not considered in test cycles PermNEDC and DoubleNEDC, both cycles require a cycle-specific conformance predicate. However, tolerance for human imprecision can be added to these predicates by means of conformance and retiming composition. Let $\mathsf{Ret}^{(1)}$ and $\mathsf{Ret}^{(2)}$ be two families of retimings. Then

$$
\begin{aligned}
\mathcal{C} &= \langle d_I,\, d_O,\, \mathsf{TraceConf}_{\epsilon_I}, && \mathsf{TraceConf}_{\epsilon_O} \rangle \\
\mathcal{C}_a &= \langle d_I,\, d_O,\, \mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_a}, && \mathsf{TraceConf}_{\epsilon_O} \rangle \\
\mathcal{C}_p &= \langle d_I,\, d_O,\, \mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_p}, && \mathsf{TraceConf}_{\epsilon_O} \rangle \\
\mathcal{C}_d &= \langle d_I,\, d_O,\, \mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_d}, && \mathsf{TraceConf}_{\epsilon_O},\, \mathsf{Sync}_d \rangle \\
\mathcal{C}(\tau_I,\epsilon_I) &= \langle d_I,\, d_O,\, \mathsf{HybridConf}_{\tau_I,\epsilon_I}, && \mathsf{TraceConf}_{\epsilon_O} \rangle \\
\mathcal{C}_p(\tau_I,\epsilon_I) &= \langle d_I,\, d_O,\, \mathsf{HybridConf}_{\tau_I,\epsilon_I} \circ \mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_p}, && \mathsf{TraceConf}_{\epsilon_O} \rangle \\
\mathcal{C}_d(\tau_I,\epsilon_I) &= \langle d_I,\, d_O,\, \mathsf{HybridConf}_{\tau_I,\epsilon_I} \circ \mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_d}, && \mathsf{TraceConf}_{\epsilon_O},\, \mathsf{Sync}_d \rangle
\end{aligned}
$$

Table 1: Overview of the evaluated contracts. For $\mathcal{C}$, $\mathcal{C}_a$, $\mathcal{C}_p$ and $\mathcal{C}_d$, $\epsilon_I = 15$ km/h. For all contracts, $\epsilon_O = 180$ mg/km, $d_I(i_1, i_2) = |i_1 - i_2|$ and $d_O(o_1, o_2) = |o_1 - o_2|$. Ret_a, Ret_p and Ret_d are defined as explained in the text above.

$\mathsf{Ret}^{(2)} \circ \mathsf{Ret}^{(1)} \triangleq \{(r_1^{(2)} \circ r_1^{(1)}, r_2^{(2)} \circ r_2^{(1)} \mid (r_1^{(2)}, r_2^{(2)}) \in \mathsf{Ret}^{(2)} \text{ and } (r_1^{(1)}, r_2^{(1)}) \in \mathsf{Ret}^{(1)}\}$ is the component-wise function composition. The definition for conformance composition is $\mathsf{Conf}_{\tau_2,\epsilon}^{\mathsf{Ret}^{(2)}} \circ \mathsf{Conf}_{\tau_1,\epsilon}^{\mathsf{Ret}^{(1)}} \triangleq \mathsf{Conf}_{\infty,\epsilon}^{\mathsf{Ret}^{(3)}}$, where $\mathsf{Ret}^{(3)} = \mathsf{Ret}_{\tau_2}^{(2)} \circ \mathsf{Ret}_{\tau_1}^{(1)}$ composes the individual retimings. The τ_1- and τ_2-constraints on $\mathsf{Ret}^{(1)}$ and $\mathsf{Ret}^{(2)}$ are applied before the composition. It is not necessary to apply further timing constraints to the resulting retiming, hence we allow infinite τ. To overcome the human imprecisions for PermNEDC and DoubleNEDC, we use the parametrised contracts $\mathcal{C}_p(\tau_I, \epsilon_I)$ and $\mathcal{C}_d(\tau_I, \epsilon_I)$, adaptations of $\mathcal{C}_p$ and $\mathcal{C}_d$, with input conformances $\mathsf{HybridPermConf}_{\tau_I,\epsilon_I} = \mathsf{HybridConf}_{\tau_I,\epsilon_I} \circ \mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_p}$ and $\mathsf{HybridDoubleConf}_{\tau_I,\epsilon_I} = \mathsf{HybridConf}_{\tau_I,\epsilon_I} \circ \mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_d}$, respectively. As for hybrid conformance in $\mathcal{C}(\tau_I, \epsilon_I)$, we will specify concrete τ_I and ϵ_I upon usage of the contracts. Notably, for DoubleNEDC, this does not have effects on the output conformance, because Sync_d does not consider the input retiming. This is important, because outputs are available only at time points 1180 and 2360 and must not be moved to time points different than that. We do not compose $\mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_a}$ and hybrid conformance, because Ret_a allows any possible NEDC permutation, which naturally reduces the effect of timing imprecisions.

Table 1 summarises the contracts presented above.

7.2. **Computing Parameters of Hybrid Conformance.** In some experiments we compute, for a fixed time threshold τ and two test cycles, the minimal value error ϵ such that hybrid conformance holds for the input. The implementation of this computation is inspired by the the HyperSTL* formula $\varphi_{\tau,\epsilon}^{\mathsf{HybridConf}}$, i.e., it computes $\min_\epsilon\ \varphi_{\tau,\epsilon}^{\mathsf{HybridConf}}$ for two test executions π_1 and π_2. The algorithm is sketched below.

$$
\begin{aligned}
\mathsf{localMin}(t_1, \mu_1, \mu_2, \tau) &= \min\, \{d_{\mathcal{Y}}(\mu_2[t_2], \mu_1[t_1]) \mid t_2 \in [t_1 - \tau; t_1 + \tau] \cap dom(\mu_2)\} \\
\mathsf{globalMin}(\mu_1, \mu_2, \tau) &= \max\, \{\mathsf{localMin}(t_1, \mu_1, \mu_2, \tau) \mid t_1 \in dom(\mu_1)\} \\
\epsilon_{\min}(\mu_1, \mu_2, \tau) &= \max\, \{\mathsf{globalMin}(\mu_1, \mu_2, \tau), \mathsf{globalMin}(\mu_2, \mu_1, \tau)\}
\end{aligned}
$$

Here, $\mathsf{localMin}(t_1, x_1, x_2, \tau)$ computes the minimal ϵ for subformula $\diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X_{\pi_1}^{*_1}) \leq \epsilon \vee \diamondsuit_{[0,\tau]}\, d_{\mathcal{Y}}(X_{\pi_2}, X_{\pi_1}^{*_1}) \leq \epsilon$, where the value of $X_{\pi_1}^{*_1}$ is frozen at time t_1. $\mathsf{globalMin}(x_1, x_2, \tau)$ reflects the *Globally* and *Freeze* operator: it finds the maximum by quantifying over all

$t_1 \in dom(x_1)$ and by calling $\mathsf{localMin}$ with the frozen time value t_1. To reflect the complete formula, $\epsilon_{\min}(x_1, x_2, \tau)$ returns the maximum of the conjuncts, which are the results of $\mathsf{globalMin}$ for both combinations of x_1 and x_2.

The computations for $\mathsf{HybridPermConf}$ and $\mathsf{HybridDoubleConf}$ proceed in two steps. Both Ret_p and Ret_d are singleton sets; it is known which retiming must be applied first. For two traces μ_1 and μ_2 and retiming (r_1, r_2), there are shifted traces $\mu_1' = \mu_1 \circ r_2$ and $\mu_2' = \mu_2 \circ r_1$. The minimal ϵ for hybrid conformance is given by $\epsilon_{\min}^{(r_1,r_2)}(\mu_1, \mu_2, \tau) = \max\{\mathsf{globalMin}(\mu_1, \mu_2', \tau), \mathsf{globalMin}(\mu_2, \mu_1', \tau)\}$.

7.3. **Test Results & Verdicts.** We executed each of NEDC, PermNEDC, DoubleNEDC and SineNEDC two times. We identify a concrete test execution by a suffix -1 or -2 to test cycle identifier (e.g., NEDC-1 is the first and NEDC-2 the second execution of NEDC). Raw data and the implementation of the analysis is available online [11]. For NEDC, we combined the result of both executions to an average value of 182 mg/km of NO_x. Notably, the Euro 6b regulation (to which our car is supposed to conform to) allows at most 80 mg/km, and the car under test is certified with 60.8 mg/km according to its documentation. The car is 3 years old.

For doping detection, a test verdict is only meaningful if its input trace is conformant to that of the average NEDC execution; otherwise, the test is trivially passed. We will first evaluate PermNEDC w.r.t. $\mathcal{C}_a$ and $\mathcal{C}_p$, DoubleNEDC w.r.t. $\mathcal{C}_d$, and SineNEDC w.r.t. $\mathcal{C}$. To demonstrate the effects of hybrid conformance, we then analyse the experiments w.r.t. the parametrised variants of the contracts $\mathcal{C}$, $\mathcal{C}_p$ and $\mathcal{C}_d$, respectively. By definition of the test cycles, the nominal value difference for PermNEDC and DoubleNEDC after retiming is zero, and for SineNEDC it is 5 km/h. Though, due to human imprecisions, the actual differences are significantly higher.

- The executions of PermNEDC are shown in Fig. 7 and 8. The amount of emitted NO_x were 392 mg/km for PermNEDC-1 and 316 mg/km for PermNEDC-2. $\mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_a}$ does hold for $\epsilon_I \geq 3$ km/h for both executions; with contract $\mathcal{C}_a$, which defines $\epsilon_I = 15$ km/h, drastic deviations of NO_x can be detected as doping. It is detected for PermNEDC-1, i.e., the cleanness test fails, as the difference of NO_x (compared to NEDC) is 210 mg/km and hence greater than $\epsilon_O = 180$ mg/km defined by $\mathcal{C}_a$. Test PermNEDC-2 passes with an NO_x difference of 134 mg/km which is within the contract.

  With contract $\mathcal{C}_p$ and input conformance $\mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_p}$, the test verdict for PermNEDC-1 is different. $\mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_p}$ would only hold for $\epsilon_I \geq 16$ km/h, which is above the contract defined threshold of 15 km/h. Hence, PermNEDC-1 is not adduced and the test trivially passed.
- DoubleNEDC-1 and 2, shown in Fig. 9 and 10, lead to an average emission of 305 mg/km, respectively 308 mg/km of NO_x. Executions of DoubleNEDC are twice as long as regular NEDC tests and produce two outputs. The measurements for DoubleNEDC-1 report $(229, 382)$ mg/km, for DoubleNEDC-2 $(207, 408)$ mg/km. To determine the verdicts for contract $\mathcal{C}_d$, we first check if $\mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_d}$ holds. This turns out not to hold for DoubleNEDC-2, because we observed value deviations of up to 25 km/h. This test is therefore trivially passed. For DoubleNEDC-1 all value deviations remain below the 15 km/h threshold; this test run is thus to be considered relevant for output comparison. According to the retiming synchronisation in $\mathcal{C}_d$, each of the outputs 229 and 382 must be compared to the NEDC output 182. The output conformance is violated for the second

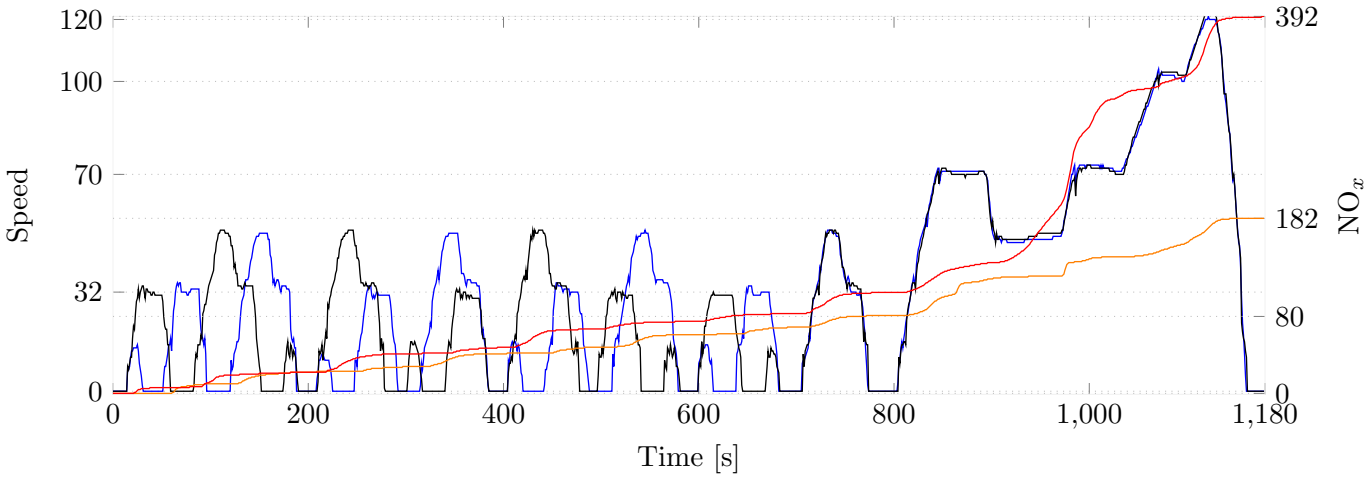

Figure 7: PermNEDC-1 speed (black) and NEDC speed (blue) in km/h, and accumulated NO_x for PermNEDC-1 (red) and NEDC (orange) in mg/km.

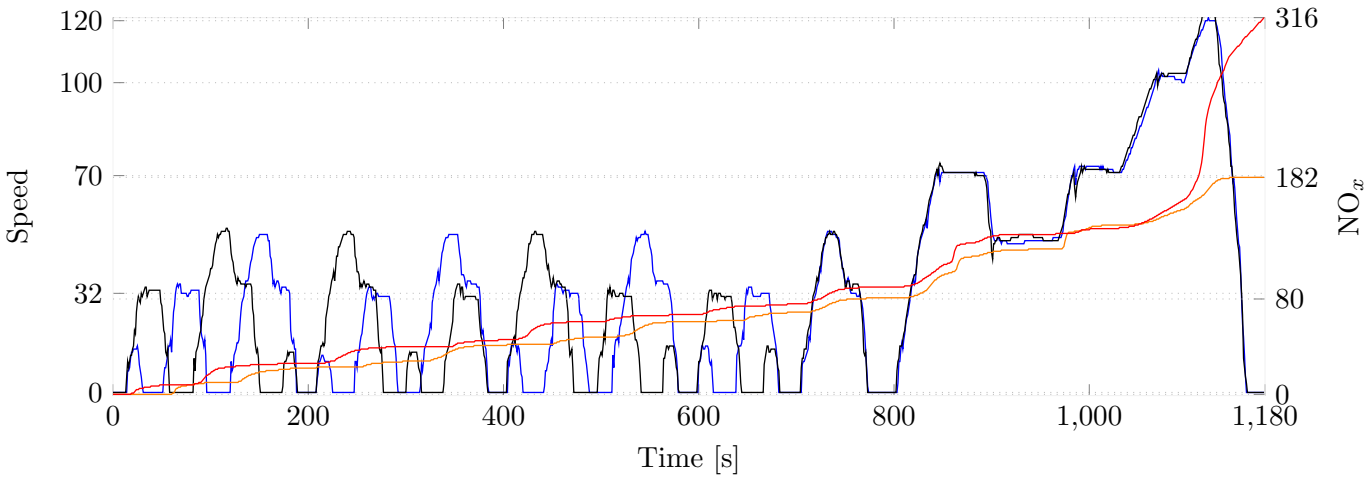

Figure 8: PermNEDC-2 speed (black) and NEDC speed (blue) in km/h, and accumulated NO_x for PermNEDC-2 (red) and NEDC (orange) in mg/km.

output, with a difference of 200 mg/km, exceeding the allowed $\epsilon_O = 180$ mg/km threshold. Hence, DoubleNEDC-1 fails — doping is detected.

- During the test executions of SineNEDC, we measured 483 mg/km and 632 mg/km. The test progression is shown in Fig. 11 and 12. In SineNEDC-1, speed values deviate by up to 18 km/h, which exceeds the ϵ_I threshold in $\mathcal{C}$, so this test run is trivially passed. SineNEDC-2 respects the ϵ_I threshold because inputs never deviate by more than 13 km/h. Consequently, SineNEDC-2 convicts our test car of doping, as the output difference of 450 mg/km is 2.5 times the allowed threshold ϵ_O.

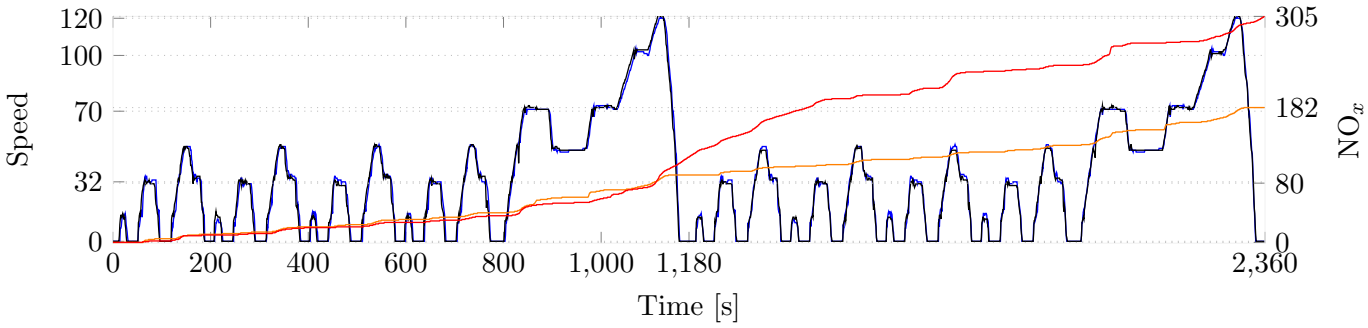

Figure 9: DoubleNEDC-1 speed (black) and NEDC speed (blue) in km/h, and accumulated NO_x for DoubleNEDC-1 (red) and NEDC (orange) in mg/km.

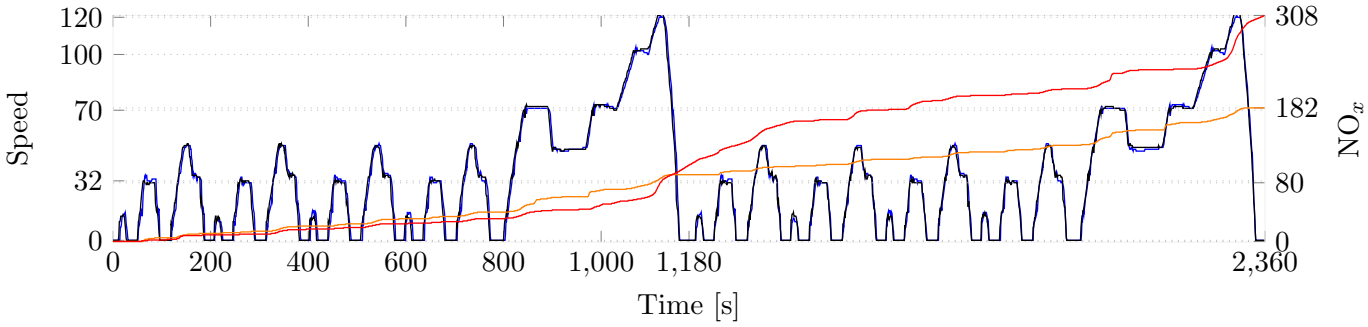

Figure 10: DoubleNEDC-2 speed (black) and NEDC speed (blue) in km/h, and accumulated NO_x for DoubleNEDC-2 (red) and NEDC (orange) in mg/km.

Test Name	*Contract*	*Input Conformance*	$\tau_I = 0$	$\tau_I = 1$	$\tau_I = 2$	$\tau_I = 3$	$\tau_I = 5$	$\tau_I = 10$	$\tau_I = 15$	$\tau_I = 20$
PermNEDC-1	$\mathcal{C}_p(\tau_I, \epsilon_I)$	HybridPermConf	$\epsilon_I = 16$	$\epsilon_I = 16$	$\epsilon_I = 16$	$\epsilon_I = 11$	$\epsilon_I = 8$	$\epsilon_I = 8$	$\epsilon_I = 8$	$\epsilon_I = 8$
PermNEDC-2	$\mathcal{C}_p(\tau_I, \epsilon_I)$	HybridPermConf	$\epsilon_I = 11$	$\epsilon_I = 10$	$\epsilon_I = 7$	$\epsilon_I = 7$	$\epsilon_I = 7$	$\epsilon_I = 7$	$\epsilon_I = 7$	$\epsilon_I = 7$
DoubleNEDC-1	$\mathcal{C}_d(\tau_I, \epsilon_I)$	HybridDoubleConf	$\epsilon_I = 15$	$\epsilon_I = 12$	$\epsilon_I = 11$	$\epsilon_I = 9$	$\epsilon_I = 6$	$\epsilon_I = 6$	$\epsilon_I = 6$	$\epsilon_I = 6$
DoubleNEDC-2	$\mathcal{C}_d(\tau_I, \epsilon_I)$	HybridDoubleConf	$\epsilon_I = 25$	$\epsilon_I = 18$	$\epsilon_I = 10$	$\epsilon_I = 8$	$\epsilon_I = 8$	$\epsilon_I = 8$	$\epsilon_I = 8$	$\epsilon_I = 8$
SineNEDC-1	$\mathcal{C}(\tau_I, \epsilon_I)$	HybridConf	$\epsilon_I = 18$	$\epsilon_I = 16$	$\epsilon_I = 15$	$\epsilon_I = 12$	$\epsilon_I = 9$	$\epsilon_I = 7$	$\epsilon_I = 6$	$\epsilon_I = 6$
SineNEDC-2	$\mathcal{C}(\tau_I, \epsilon_I)$	HybridConf	$\epsilon_I = 13$	$\epsilon_I = 11$	$\epsilon_I = 9$	$\epsilon_I = 9$	$\epsilon_I = 7$	$\epsilon_I = 7$	$\epsilon_I = 7$	$\epsilon_I = 7$

Table 2: Comparison of minimal value thresholds ϵ_I for fixed τ_I. Values are given as km/h and time in seconds.

- As discussed, we use hybrid conformance to compensate for human driving imprecisions. In this context, Table 2 details the effect of a choice of τ on the maximal value error. We fix a maximum value that we allow for the time offset τ_I. For this τ_I we analyse our dataset to find the minimal ϵ_I such that for the combination of τ_I and ϵ_I the input traces under consideration satisfy the cycle-specific hybrid conformance. For $\tau_I = 0$ we get exactly the ϵ_I for which the two traces satisfy $\mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_p}$ (for PermNEDC), $\mathsf{Conf}_{\epsilon_I}^{\mathsf{Ret}_d}$ (for DoubleNEDC), and $\mathsf{TraceConf}_{\epsilon_O}$ (for SineNEDC). Table 2 shows the computed ϵ_I values for $\tau_I = 0, 1, 2, 5, 10, 15$ and 20 seconds. As expected, an increasing τ_I induces the

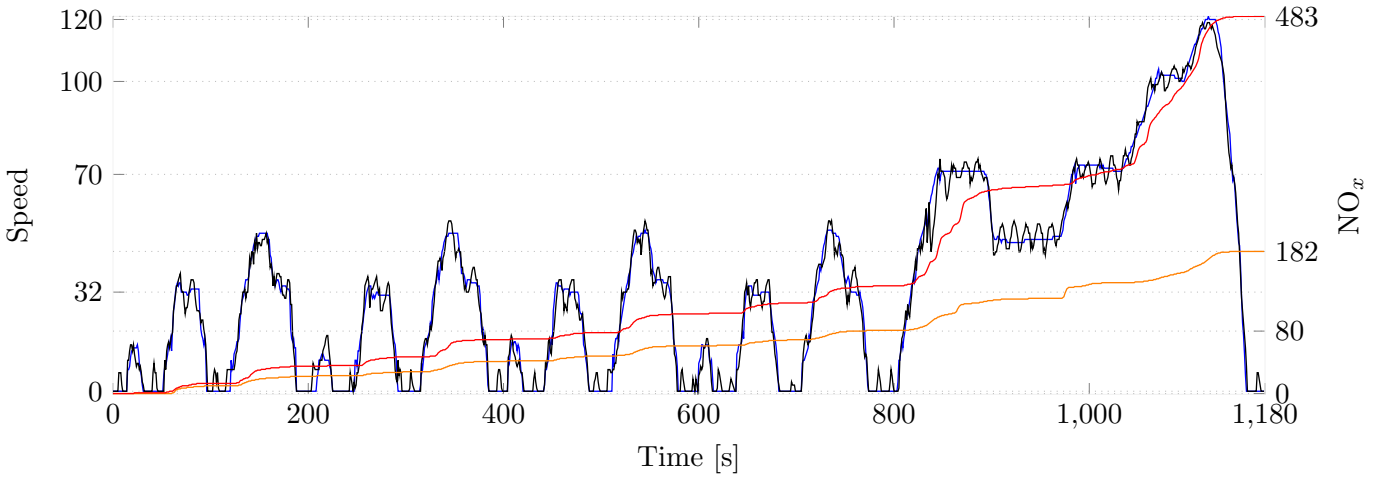

Figure 11: SineNEDC-1 speed (black) and NEDC speed (blue) in km/h, and accumulated NO_x for SineNEDC-1 (red) and NEDC (orange) in mg/km.

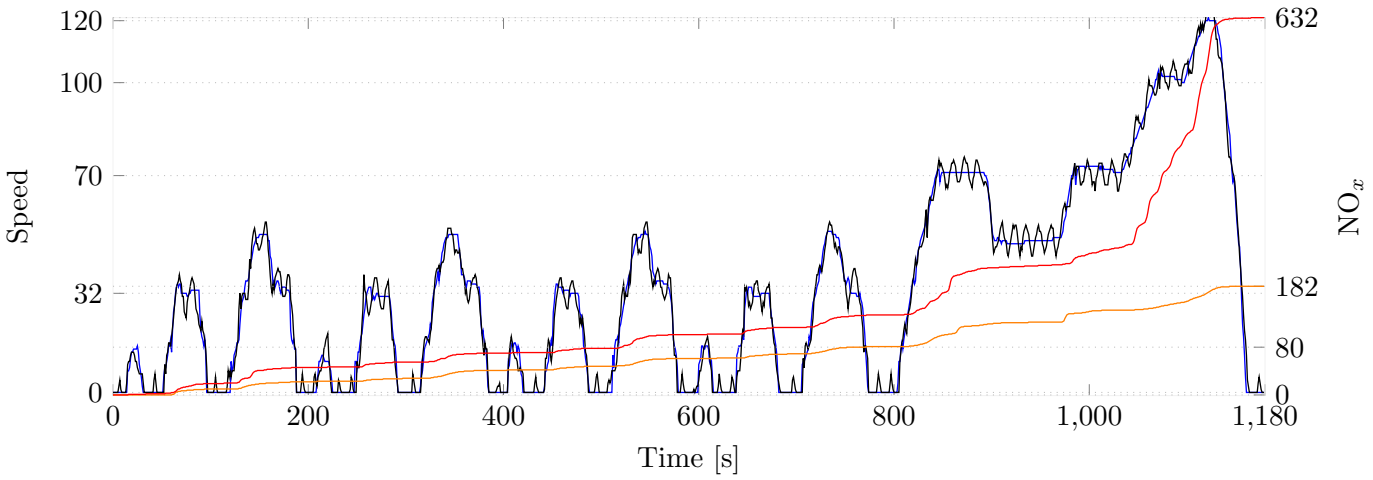

Figure 12: SineNEDC-2 speed (black) and NEDC speed (blue) in km/h, and accumulated NO_x for SineNEDC-2 (red) and NEDC (orange) in mg/km.

minimal ϵ_I to decrease. At $\tau_I = 5$ the decrease in the value error reduces notably. This happens because the error is only partially caused by the incorrect timing of the driver. From the values reported in Table 2 we see that if we allow deviation for the input $\tau_I = 2$, and keep $\epsilon_I = 15$, then we have that $\mathsf{HybridDoubleConf}_{\tau_I,\epsilon_I}(\text{NEDC}, \text{DoubleNEDC-2})$ and $\mathsf{HybridConf}_{\tau_I,\epsilon_I}(\text{NEDC}, \text{SineNEDC-1})$ hold. For time threshold $\tau_I = 3$ seconds $\mathsf{HybridPermConf}_{\tau_I,\epsilon_I}(\text{NEDC}, \text{PermNEDC-1})$ also holds. Thus, under hybrid conformance these pairs of traces will be considered in the cleanness test for contracts $\mathcal{C}_d(2, 15)$, $\mathcal{C}(2, 15)$ and $\mathcal{C}_p(3, 15)$, respectively, while under their original contract and input conformance they are to be dismissed.

7.4. **Evaluation and Discussion.** The amounts of emitted NO_x observed during our experiments provide clear indications of software doping regarding the car's emission cleaning system. The conformance-based contracts provide the formal basis for this verdict, as discussed above. We here complement this fact with a more intuitive explanation of the behaviour observed.

- PermNEDC slightly reorders NEDC segments in the UDC part of the test cycle. During this part, the measured NO_x does not significantly differ from the NEDC reference. However, during the (unmodified) EUDC part, the amount of emissions grows significantly. It is very unlikely to find a physical explanation for the NO_x increase; and very likely, that the cleaning system is optimised specifically for the NEDC.
- The DoubleNEDC executions appear to reveal that the emission cleaning system optimisation can also rely on engine temperature or execution time instead of speed data. Physically, many of the common emission cleaning techniques require a hot engine to work properly (and none of them requires a cold engine). Therefore, a lower NO_x value can be expected if the NEDC is run with a hot engine. In our experiments, however, the NO_x emissions in the hot half are almost two times higher than in the initial cold part. In other words, the emission cleaning performance is reduced after the first NEDC execution. There is no physical explanation for this behaviour. Inside the software, detecting the end of an NEDC trip can be implemented very easily, for instance with a timer counting from 1180 — the length of NEDC — to zero.
- With SineNEDC, we test the cleaning system during driving behaviour which is rich in accelerations and decelerations. An increased amount of NO_x can possibly be explained by physical phenomena. However, we measured an increase of factors 2.7 and 3.5; these numbers can be safely considered as too high for a trustworthy emission cleaning system.

Software doping theory provides the basis for detecting software behaviour violating a formal contract. In this, physical aspects of the emission cleaning system should be considered during the construction of test cases, and test cycles for which drastically higher emissions can be explained physically, should not be considered. The test cycles we used for our experiments were picked with automotive expertise to avoid physically stressful cycles. If test cases are generated automatically from a contract, the physical constraints could be captured by the contract.

The contracts we use for our experiments can be interpreted as very generous in favour of the manufacturers. Input thresholds such as 15 km/h and 2 seconds appear as reasonable values, keeping all tests close enough to the original NEDC. For the output threshold, we use a very large deviation value of 180 mg/km, which allows NO_x emissions to almost double compared to the original NEDC value. Despite the generosity of the contracts, our experiments have been able to reveal doping for all experiments except PermNEDC-2.

The analysis of the data shows that it is indeed necessary to not only consider a deviation of value, but to also allow for timing deviations. Considering value and timing deviations offers a rich set of potential test cycles for doping tests and allows to realistically verify conformance of a test cycle and a reference cycle; especially when the quality of the studied driving tests suffers from the human-caused input distortions. In this regard, cleanness notions entailing hybrid conformance are more adequate than conformance notions demanding punctual test executions, such as robust cleanness. Without hybrid conformance, more of the doping cases we have detected would slip through.

Finally, while hybrid conformance is central to the case study considered here, our generic theory of conformance-based cleanness allows for using other conformance notions as appropriate for the CPS under test.

Acknowledgments

The work of Sebastian Biewer and Holger Hermanns is supported by the ERC Grant 695614 (POWVER) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) grant 389792660 as part of TRR 248, see `https://perspicuous-computing.science`. The work of Sebastian Biewer is supported by the Saarbrücken Graduate School of Computer Science. The work of Holger Hermanns is supported by the Key-Area Research and Development Program Grant 2018B010107004 of Guangdong Province. The work of Mohammad Reza Mousavi has been partially supported by the UKRI Trustworthy Autonomous Systems Node in Verifiability, with Grant Award Reference EP/V026801/1.

References

[1] H. Abbas, B. Hoxha, G. E. Fainekos, J. V. Deshmukh, J. Kapinski, and K. Ueda. Wip abstract: Conformance testing as falsification for cyber-physical systems. In *ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS, Berlin, Germany, April 14-17, 2014*, page 211. IEEE Computer Society, 2014.

[2] H. Abbas, H. D. Mittelmann, and G. E. Fainekos. Formal property verification in a conformance testing framework. In *Twelfth ACM/IEEE International Conference on Formal Methods and Models for Codesign, MEMOCODE 2014, Lausanne, Switzerland, October 19-21, 2014*, pages 155–164. IEEE, 2014.

[3] A. Aerts, M. Reniers, and M. Mousavi. Chapter 19 - model-based testing of cyber-physical systems. In H. Song, D. B. Rawat, S. Jeschke, and C. Brecher, editors, *Cyber-Physical Systems*, Intelligent Data-Centric Systems, pages 287 – 304. Academic Press, Boston, 2017.

[4] S. Agrawal and B. Bonakdarpour. Runtime verification of k-safety hyperproperties in hyperltl. In *IEEE 29th Computer Security Foundations Symposium, CSF 2016, Lisbon, Portugal, June 27 - July 1, 2016*, pages 239–252. IEEE Computer Society, 2016.

[5] R. Alur and T. A. Henzinger. A really temporal logic. *J. ACM*, 41(1):181–204, 1994.

[6] Y. Annpureddy, C. Liu, G. E. Fainekos, and S. Sankaranarayanan. S-taliro: A tool for temporal logic falsification for hybrid systems. In P. A. Abdulla and K. R. M. Leino, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 17th International Conference, TACAS 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26-April 3, 2011. Proceedings*, volume 6605 of *Lecture Notes in Computer Science*, pages 254–257. Springer, 2011.

[7] H. L. S. Araujo, G. Carvalho, M. Mohaqeqi, M. R. Mousavi, and A. Sampaio. Sound conformance testing for cyber-physical systems: Theory and implementation. *Sci. Comput. Program.*, 162:35–54, 2018.

[8] G. Barthe, P. R. D'Argenio, B. Finkbeiner, and H. Hermanns. Facets of software doping. In T. Margaria and B. Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications - 7th International Symposium, ISoLA 2016, Part II*, volume 9953 of *LNCS*, pages 601–608, 2016.

[9] S. Biewer, P. R. D'Argenio, and H. Hermanns. Cyber-physical doping tests. In *3rd Workshop on Monitoring and Testing of Cyber-Physical Systems, MT@CPSWeek 2018, Porto, Portugal, April 10, 2018*, pages 18–19. IEEE, 2018.

[10] S. Biewer, P. R. D'Argenio, and H. Hermanns. Doping tests for cyber-physical systems. In D. Parker and V. Wolf, editors, *Quantitative Evaluation of Systems, 16th International Conference, QEST 2019, Glasgow, UK, September 10-12, 2019, Proceedings*, volume 11785 of *Lecture Notes in Computer Science*, pages 313–331. Springer, 2019.

[11] S. Biewer, M. Fries, and T. Heinze. Conformance relations and hyperproperties for doping detection in time and space (supplementary material). `https://www.powver.org/publications/conformance-based-doping-detection`, 2020.

[12] B. Bonakdarpour, P. Prabhakar, and C. Sánchez. Model checking timed hyperproperties in discrete-time systems. In R. Lee, S. Jha, and A. Mavridou, editors, *NASA Formal Methods - 12th International Symposium, NFM 2020, Moffett Field, CA, USA, May 11-15, 2020, Proceedings*, volume 12229 of *Lecture Notes in Computer Science*, pages 311–328. Springer, 2020.

[13] N. Brett, U. Siddique, and B. Bonakdarpour. Rewriting-based runtime verification for alternation-free hyperltl. In A. Legay and T. Margaria, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 23rd International Conference, TACAS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings, Part II*, volume 10206 of *Lecture Notes in Computer Science*, pages 77–93, 2017.

[14] L. Brim, P. Dluhos, D. Safranek, and T. Vejpustek. STL*: Extending signal temporal logic with signal-value freezing operator. *Inf. Comput.*, 236:52–67, 2014.

[15] M. R. Clarkson, B. Finkbeiner, M. Koleini, K. K. Micinski, M. N. Rabe, and C. Sánchez. Temporal logics for hyperproperties. In M. Abadi and S. Kremer, editors, *POST 2014*, volume 8414 of *LNCS*, pages 265–284. Springer, 2014.

[16] M. R. Clarkson and F. B. Schneider. Hyperproperties. In *CSF'08*, pages 51–65, 2008.

[17] M. Contag, G. Li, A. Pawlowski, F. Domke, K. Levchenko, T. Holz, and S. Savage. How they did it: An analysis of emission defeat devices in modern automobiles. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 231–250. IEEE Computer Society, 2017.

[18] P. R. D'Argenio, G. Barthe, S. Biewer, B. Finkbeiner, and H. Hermanns. Is your software on dope? - formal analysis of surreptitiously "enhanced" programs. In H. Yang, editor, *Programming Languages and Systems - 26th European Symposium on Programming, ESOP 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, volume 10201 of *Lecture Notes in Computer Science*, pages 83–110. Springer, 2017.

[19] R. De Nicola and M. Hennessy. Testing equivalences for processes. *Theor. Comput. Sci.*, 34:83–133, 1984.

[20] S. Demri and R. Lazic. LTL with the freeze quantifier and register automata. *ACM Trans. Comput. Log.*, 10(3):16:1–16:30, 2009.

[21] J. V. Deshmukh, R. Majumdar, and V. S. Prabhu. Quantifying conformance using the skorokhod metric. *Formal Methods in System Design*, 50(2-3):168–206, 2017.

[22] R. Dimitrova, M. Gazda, M. R. Mousavi, S. Biewer, and H. Hermanns. Conformance-based doping detection for cyber-physical systems. In A. Gotsman and A. Sokolova, editors, *Formal Techniques for Distributed Objects, Components, and Systems - 40th IFIP WG 6.1 International Conference, FORTE 2020, Held as Part of the 15th International Federated Conference on Distributed Computing Techniques, DisCoTec 2020, Valletta, Malta, June 15-19, 2020, Proceedings*, volume 12136 of *Lecture Notes in Computer Science*, pages 59–77. Springer, 2020.

[23] A. Donzé. Breach, A toolbox for verification and parameter synthesis of hybrid systems. In T. Touili, B. Cook, and P. B. Jackson, editors, *Computer Aided Verification, 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings*, volume 6174 of *Lecture Notes in Computer Science*, pages 167–170. Springer, 2010.

[24] G. E. Fainekos and G. J. Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theor. Comput. Sci.*, 410(42):4262–4291, 2009.

[25] B. Finkbeiner and C. Hahn. Deciding hyperproperties. In J. Desharnais and R. Jagadeesan, editors, *27th International Conference on Concurrency Theory, CONCUR 2016, August 23-26, 2016, Québec City, Canada*, volume 59 of *LIPIcs*, pages 13:1–13:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[26] B. Finkbeiner, C. Hahn, and M. Stenger. EAHyper: Satisfiability, implication, and equivalence checking of hyperproperties. In R. Majumdar and V. Kuncak, editors, *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part II*, volume 10427 of *Lecture Notes in Computer Science*, pages 564–570. Springer, 2017.

[27] B. Finkbeiner, C. Hahn, M. Stenger, and L. Tentrup. Monitoring hyperproperties. In S. K. Lahiri and G. Reger, editors, *Runtime Verification - 17th International Conference, RV 2017, Seattle, WA, USA, September 13-16, 2017, Proceedings*, volume 10548 of *Lecture Notes in Computer Science*, pages 190–207. Springer, 2017.

[28] B. Finkbeiner, C. Hahn, M. Stenger, and L. Tentrup. RVHyper: A runtime verification tool for temporal hyperproperties. In D. Beyer and M. Huisman, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 24th International Conference, TACAS 2018, Held as Part of the European Joint*

Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings, Part II, volume 10806 of *Lecture Notes in Computer Science*, pages 194–200. Springer, 2018.

[29] B. Finkbeiner, C. Hahn, M. Stenger, and L. Tentrup. Monitoring hyperproperties. *Formal Methods Syst. Des.*, 54(3):336–363, 2019.

[30] B. Finkbeiner, M. N. Rabe, and C. Sánchez. Algorithms for model checking HyperLTL and HyperCTL*. In D. Kroening and C. S. Pasareanu, editors, *CAV 2015*, volume 9206 of *LNCS*, pages 30–48. Springer, 2015.

[31] M. Gazda and M. R. Mousavi. Logical characterisation of hybrid conformance. In A. Czumaj, A. Dawar, and E. Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 130:1–130:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[32] A. Girard, A. A. Julius, and G. J. Pappas. Approximate simulation relations for hybrid systems. *Discrete Event Dynamic Systems*, 18(2):163–179, 2008.

[33] A. Girard and G. J. Pappas. Approximate bisimulation: A bridge between computer science and control theory. *Eur. J. Control*, 17(5-6):568–578, 2011.

[34] C. Hahn, M. Stenger, and L. Tentrup. Constraint-based monitoring of hyperproperties. In T. Vojnar and L. Zhang, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 25th International Conference, TACAS 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings, Part II*, volume 11428 of *Lecture Notes in Computer Science*, pages 115–131. Springer, 2019.

[35] T. Hapke, P. Hornung, and J. Becker. Schummeln auch in Europa. ARD / Norddeutscher Rundfunk, `https://www.tagesschau.de/wirtschaft/vw-schummelsoftware-101.html`, 2015. Online; accessed: 2019-04-19.

[36] F. C. Hennie. Fault detecting experiments for sequential circuits. In *5th Annual Symposium on Switching Circuit Theory and Logical Design, Princeton, New Jersey, USA, November 11-13, 1964*, pages 95–110. IEEE Computer Society, 1964.

[37] H. Hermanns, S. Biewer, P. R. D'Argenio, and M. A. Köhl. Verification, testing, and runtime monitoring of automotive exhaust emissions. In G. Barthe, G. Sutcliffe, and M. Veanes, editors, *LPAR-22. 22nd International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Awassa, Ethiopia, 16-21 November 2018*, volume 57 of *EPiC Series in Computing*, pages 1–17. EasyChair, 2018.

[38] H. Ho, R. Zhou, and T. M. Jones. On verifying timed hyperproperties. In J. Gamper, S. Pinchinat, and G. Sciavicco, editors, *26th International Symposium on Temporal Representation and Reasoning, TIME 2019, October 16-19, 2019, Málaga, Spain*, volume 147 of *LIPIcs*, pages 20:1–20:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[39] N. Khakpour and M. R. Mousavi. Notions of conformance testing for cyber-physical systems: Overview and roadmap (invited paper). In L. Aceto and D. de Frutos-Escrig, editors, *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015*, volume 42 of *LIPIcs*, pages 18–40. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[40] M. A. Köhl, H. Hermanns, and S. Biewer. Efficient monitoring of real driving emissions. In C. Colombo and M. Leucker, editors, *Runtime Verification - 18th International Conference, RV 2018, Limassol, Cyprus, November 10-13, 2018, Proceedings*, volume 11237 of *Lecture Notes in Computer Science*, pages 299–315. Springer, 2018.

[41] D. Lee and M. Yannakakis. Principles and methods of testing finite-state machines - a survey. *Proceedings of the IEEE*, 84(8):1089–1123, 1996.

[42] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In Y. Lakhnech and S. Yovine, editors, *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Joint International Conferences on Formal Modelling and Analysis of Timed Systems, FORMATS 2004 and Formal Techniques in Real-Time and Fault-Tolerant Systems, FTRTFT 2004, Grenoble, France, September 22-24, 2004, Proceedings*, volume 3253 of *Lecture Notes in Computer Science*, pages 152–166. Springer, 2004.

[43] L. V. Nguyen, J. Kapinski, X. Jin, J. V. Deshmukh, and T. T. Johnson. Hyperproperties of real-valued signals. In J. Talpin, P. Derler, and K. Schneider, editors, *Proceedings of the 15th ACM-IEEE International Conference on Formal Methods and Models for System Design, MEMOCODE 2017, Vienna, Austria, September 29 - October 02, 2017*, pages 104–113. ACM, 2017.

[44] A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 46–57. IEEE Computer Society, 1977.

[45] D. Safránek, M. Troják, V. Bruza, T. Vejpustek, J. Papousek, M. Demko, S. Pastva, A. Pejznoch, and L. Brim. Barbaric robustness monitoring revisited for stl* in parasim. In L. Bortolussi and G. Sanguinetti, editors, *Computational Methods in Systems Biology - 17th International Conference, CMSB 2019, Trieste, Italy, September 18-20, 2019, Proceedings*, volume 11773 of *Lecture Notes in Computer Science*, pages 356–359. Springer, 2019.

[46] J. Tretmans. *A formal Approach to conformance testing*. PhD thesis, University of Twente, The Netherlands, 1992.

[47] J. Tretmans. Conformance testing with labelled transition systems: Implementation relations and test generation. *Computer Networks and ISDN Systems*, 29(1):49–79, 1996.

[48] United Nations. UN Vehicle Regulations - 1958 Agreement, Revision 2, Addendum 100, Regulation No. 101, Revision 3 — E/ECE/324/Rev.2/Add.100/Rev.3, 2013.

[49] M. van Osch. Hybrid input-output conformance and test generation. In K. Havelund, M. Núñez, G. Rosu, and B. Wolff, editors, *Formal Approaches to Software Testing and Runtime Verification, First Combined International Workshops, FATES 2006 and RV 2006, Seattle, WA, USA, August 15-16, 2006, Revised Selected Papers*, volume 4262 of *Lecture Notes in Computer Science*, pages 70–84. Springer, 2006.