Form Methods Syst Des (2017) 50:168–206
DOI 10.1007/s10703-016-0261-8

Quantifying conformance using the Skorokhod metric

Jyotirmoy V. Deshmukh[1] · **Rupak Majumdar**[2] · **Vinayak S. Prabhu**[2,3]

Published online: 12 April 2017

Abstract The conformance testing problem for dynamical systems asks, given two dynamical models (e.g., as Simulink diagrams), whether their behaviors are "close" to each other. In the semi-formal approach to conformance testing, the two systems are simulated on a large set of tests, and a metric, defined on pairs of real-valued, real-timed trajectories, is used to determine a lower bound on the distance. We show how the Skorokhod metric on continuous dynamical systems can be used as the foundation for conformance testing of complex dynamical models. The Skorokhod metric allows for both state value mismatches and timing distortions, and is thus well suited for checking conformance between idealized models of dynamical systems and their implementations. We demonstrate the robustness of the metric by proving a *transference theorem*: trajectories close under the Skorokhod metric satisfy "close" logical properties in the timed linear time logic FLTL (Freeze LTL) containing a rich class of temporal and spatial constraint predicates involving time and value freeze variables. We provide efficient window-based streaming algorithms to compute the Skorokhod metric for both piecewise affine and piecewise constant traces, and use these as a basis for a conformance testing tool for Simulink. We experimentally demonstrate the effectiveness of our tool in finding discrepant behaviors on a set of control system benchmarks, including an industrial challenge problem.

Keywords Trace metrics · Conformance testing · Signal temporal logic (STL) · Skorokhod metric · Timing distortion

✉ Vinayak S. Prabhu
vinayak@mpi-sws.org

Jyotirmoy V. Deshmukh
jyotirmoy.deshmukh@tema.toyota.com

Rupak Majumdar
rupak@mpi-sws.org

[1] Toyota Technical Center, Ann Arbor, MI, USA

[2] MPI-SWS, Kaiserslautern, Germany

[3] University of Porto, Porto, Portugal

1 Introduction

A fundamental question in model-based design is *conformance testing*: whether two models of a system display similar behavior. For discrete systems, this question is well-studied [21, 22,31,32], and there is a rich theory of process equivalences based, e.g., on bisimilarity. For continuous and hybrid systems, however, the state of the art is somewhat unsatisfactory. While there is a straightforward generalization of process equivalences to the continuous case, in practice, equivalence notions such as bisimilarity are always too strong and most systems are not bisimilar. Since equivalence is a Boolean notion, one gets no additional information about the systems other than they are "not bisimilar." Further, even if two dynamical systems are bisimilar, they may still differ in many control-theoretic properties. Thus, classical notions for equivalence and conformance have been of limited use in industrial practice.

In recent years, the notion of bisimulation has therefore been generalized to *metrics* on systems, which quantify the distance between them. For example, one approach is that of ϵ-bisimulation, which requires that the states of the two systems remain "close" forever (within an ϵ-ball), rather than coincide exactly. Under suitable stability assumptions on the dynamics, one can construct ϵ-bisimulations [19,20]. Unfortunately, proving the pre-requisites for the existence of ϵ-bisimulations for complex dynamical models, or coming up with suitable and practically tractable bisimulation functions is extremely difficult in practice. In addition, establishing ϵ-bisimulation requires full knowledge of the system dynamics making the scheme inapplicable where one system is an actual physical component with unknown dynamics. So, these notions have also been of limited industrial use so far.

Instead, a more pragmatic semi-formal approach has gained prominence in industrial practice. In this approach, the two systems are executed on the same input sequences and a metric on finite trajectories is used to evaluate the closeness of these trajectories. The key to this methodology is the selection of a *good* metric, with the following properties:

- *Transference* Closeness in the metric must translate to preserving interesting classes of logical and functional specifications between systems, and
- *Tractability* The metric should be efficiently computable.

In addition, there is the more informal requirement of *usability*: the metric should classify systems which the engineers consider close as being close, and conversely.

The simplest candidate metric is a *pointwise* metric that computes the maximum pointwise difference between two trajectories, sometimes generalized to apply a constant time-shift to one trajectory [16]. Unfortunately, for many practical models, two trajectories may be close only under variable time-shifts. This is the case, for example, for two dynamical models that may use different numerical integration techniques (e.g., fixed step versus adaptive step) or when some component in the implementation has some jitter. Thus, the pointwise metric spuriously reports large distances for "close" models. More nuanced hybrid distances have been proposed [1], but the transference properties of these metrics w.r.t. common temporal logics are not yet clear.

In this work we present a methodology for quantifying conformance between real-valued dynamical systems based on the *Skorokhod* metric [14]. The Skorokhod metric allows for mismatches in both the trace values and in the timeline, and quantifies temporal and spatial variation of the system dynamics under a unifying framework. The distortion of the timeline is specified by a *retiming* function r which is a continuous bijective strictly increasing function from $\mathbb{R}_+$ to $\mathbb{R}_+$. Using the retiming function, we obtain the *retimed trace* $x\,(\mathsf{r}(t))$ from the original trace $x(t)$. Intuitively, in the retimed trace $x\,(\mathsf{r}(t))$, we see exactly the same values as before, in exactly the same order, but the time duration between two values might now be

different than the corresponding duration in the original trace. The amount of distortion for the retiming r is given by $\sup_{t\geq 0} |\mathsf{r}(t) - t|$. Using retiming functions, the Skorokhod distance between two traces x and y is defined to be the least value over all possible retimings r of:

$$\max\left(\sup_{t\in[0,T]} |\mathsf{r}(t) - t|\,,\ \sup_{t\in[0,T]} \mathcal{D}_{\mathcal{O}}\left(x\left(\mathsf{r}(t)\right), y(t)\right)\right),$$

where $\mathcal{D}_{\mathcal{O}}$ is a pointwise metric on values. In this formula, the first component quantifies the *timing discrepancy* of the timing distortion required to "match" two traces, and the second quantifies the *value mismatch* (in the metric space $\mathcal{O}$) of the values under the timing distortion. The Skorokhod metric was introduced as a theoretical basis for defining the semantics of hybrid systems by providing an appropriate hybrid topology [10,11]. We now demonstrate its usefulness in the context of conformance testing.

Transference We show that the Skorokhod metric gives a robust quantification of system conformance by relating the metric to TLTL (timed LTL) in the boolean setting; and to FLTL (Freeze LTL) in the real-valued signal setting. The logics contain (i) predicates of the form $f(x_1, \ldots, x_n) \geq 0$, as in Signal Temporal Logic (STL) [16] (we however allow f to be non-linear), for specifying constraints on trace values; and (ii) *freeze quantifiers*, as in TPTL [4] and STL* [8,9][1] for specifying temporal and value constraints relating different parts of traces (freeze quantifiers can express more complex timing constraints than bounded timing constraints, e.g., of MTL, and can also specify complex value constraints such as signal tracking which cannot be expressed by STL). TLTL subsumes MTL in the boolean setting; and FLTL subsumes STL in the real-valued setting. We prove a *transference theorem*: flows (and propositional traces) which are close under the Skorokhod metric satisfy "close" FLTL (resp. TLTL) formulae for a rich class of temporal and spatial predicates, where the untimed structure of the formulae remains unchanged, only the predicates are enlarged.

Tractability We improve on recent polynomial-time algorithms for the Skorokhod metric [27] between polygonal (piecewise affine and continuous) traces by taking advantage of the fact that, in practice, only retimings that map the times in one trace to "close" times in the other are of interest. This enables us to obtain a streaming sliding-window based monitoring procedure which takes only $O(W)$ time per sample, where W is the window size (assuming the dimension n of the system to be a constant). In this work, we also develop and implement a significantly faster sliding-window based Skorokhod metric computation procedure for *piecewise constant* traces, and experimentally compare the tradeoff between faster computation time of the piecewise constant routine and the discrepancy compared to the polygonal trace procedure.

Usability Using the Skorokhod distance checking procedure as a subroutine, we have implemented a Simulink toolbox for conformance testing. Our tool integrates with Simulink's model-based design flow for control systems, and provides a stochastic search-based approach to find inputs which maximize the Skorokhod distance between systems under these inputs.

We present three case studies from the control domain, including industrial challenge problems; our empirical evaluation shows that our tool computes sharp estimates of the conformance distance reasonably fast on each of them. Our input models were complex enough that techniques such as ϵ-bisimulation functions are inapplicable. We conclude that the Skorokhod metric can be an effective foundation for semi-formal conformance testing for complex dynamical models.

[1] TPTL has only time freeze variables, and STL * has only value freeze variables. FLTL has both. Moreover, the predicates over the freeze variables can be non-linear in FLTL, unlike STL * which only allows linear predicates.

Related work The work of [1,2] is closely related to ours. In it, robustness properties of hybrid state sequences are derived with respect to a trace metric which also quantifies temporal and spatial variations. Our work differs in the following ways. First, we guarantee robustness properties over *flows* rather than only over (discrete) sequences. Second, the Skorokhod metric is a stronger form of the $(T, J, (\tau, \epsilon))$-closeness degree[2,3](for systems which do not have hybrid time); and allows us to give stronger robustness transference guarantees. The Skorokhod metric requires order preservation of the timeline, which the $(T, J, (\tau, \epsilon))$-closeness function does not. Preservation of the timeline order allows us to (i) keep the untimed structure of the formulae the same (unlike in the transference theorem of [1]); (ii) show transference of a rich class of global timing constraints using freeze quantifiers (rather than only for the standard bounded time quantifiers of MTL/MITL). However, for implementations where the timeline order is not preserved, we have to settle for the less stronger guarantees provided by [1]. The work of [16] deals with spatial robustness of STL; the only temporal disturbances considered are constant time-shifts for the entire signal where the entire signal is moved to the past, or to the future by the same amount. In contrast, the Skorokhod metric incorporates variable time-shifts.

Summary of Results The present work contains the following results.

1. We develop an efficient algorithm for computing the Skorokhod distance between piecewise constant traces (for a general metric space) that may have discontinuities.
2. We implement the above piecewise constant trace algorithm for $\mathbb{R}^n$, and also the polygonal trace algorithm from [27], and compare the two implementations. Our piecewise constant trace algorithm implementation runs two orders of magnitude faster than the polygonal trace algorithm procedure. We also experimentally explore the tradeoff between faster computation time of the piecewise constant routine and the distance values obtained compared to the polygonal procedure.
3. We introduce the expressive logic FLTL (based on the logic STL* ([8,9]), which augments LTL with *value* and *time* freeze variables, and general constraint predicates. We prove a logic transference result with respect to this logic: flows which are close under the Skorokhod metric satisfy "close" FLTL formulae. Value and time freeze variables allow expression of complex properties of traces, such as signal tracking ([8,9]).
4. We present detailed examples explaining and demonstrating our logic transference guarantees.
5. We present a new application of the Skorokhod metric: we develop an algorithm to quantify the *timing distortion* between traces under some allowed value distortion. More precisely, given polygonal (or piecewise constant) traces x, y, and an $\epsilon \geq 0$, we develop an algorithm to compute the minimal retiming r required in order that $x\,(r(t))$ and $y(t)$ match, modulo ϵ, for all t. More precisely, we compute the retiming which minimizes the quantity $\sup_{t\in[0,T]} |\mathsf{r}(t) - t|$ (this value quantifies the timing distortion under r) such that $\sup_{t\in[0,T]} \mathcal{D}\left(x\left(r(t)\right),\ y(t)\right) \leq \epsilon$. We also implement the algorithm and present experimental results.

[2] Instead of having only two parameters τ and ϵ for time and state variation, we generalize to $n+1$ parameters: we pre-scale time and the n state components with $n+1$ constants, and have a single value quantifying closeness of the scaled traces.

[3] Informally, two signals x, y are $(T, J, (\tau, \epsilon))$-close if for each point $x(t)$, there is a point $y(t')$ with $|t-t'| < \tau$ such that $\mathcal{D}(x(t), y(t')) < \epsilon$; and similarly for $y(t)$.

2 Conformance testing with the Skorokhod metric

2.1 Systems and conformance testing

Traces and systems A (finite) *trace* or a *signal* $\pi : [T_i, T_e] \to \mathcal{O}$ is a mapping from a finite closed interval $[T_i, T_e]$ of $\mathbb{R}_+$, with $0 \leq T_i < T_e$, to some topological space $\mathcal{O}$. If $\mathcal{O}$ is a metric space, we refer to the associated metric on $\mathcal{O}$ as $\mathcal{D}_{\mathcal{O}}$. The time-domain of π, denoted $\mathsf{tdom}(\pi)$, is the time interval $[T_i, T_e]$ over which it is defined. The time-duration of π, denoted $\mathsf{tlen}(\pi)$, is $T_e - T_i$. The t-suffix of π for $t \in \mathsf{tdom}(\pi)$, denoted π^t, is the trace π restricted to the interval $[t, \max \mathsf{tdom}(\pi)]$. We denote by $\pi_{\downarrow T_e'}$ the prefix trace obtained from π by restricting the domain to $[T_i, T_e'] \subseteq \mathsf{tdom}(\pi)$.

A (continuous-time) *system* $\mathfrak{A} : \left(\mathbb{R}_+^{[]} \mapsto \mathcal{O}_{\mathsf{ip}}\right) \to \left(\mathbb{R}_+^{[]} \to \mathcal{O}_{\mathsf{op}}\right)$, where $\mathbb{R}_+^{[]}$ is the set of finite closed intervals of $\mathbb{R}_+$, transforms input traces $\pi_{\mathsf{ip}} : [T_i, T_e] \to \mathcal{O}_{\mathsf{ip}}$ into output traces $\pi_{\mathsf{op}} : [T_i, T_e] \to \mathcal{O}_{\mathsf{op}}$ (over the same time domain). We require that the system is *causal*: if $\mathfrak{A}(\pi_{\mathsf{ip}}) \to \pi_{\mathsf{op}}$, then for every $\min \mathsf{tdom}(\pi) \leq T_e' < \max \mathsf{tdom}(\pi)$, the system $\mathfrak{A}$ maps $\pi_{\mathsf{ip}\downarrow T_e'}$ to $\pi_{\mathsf{op}\downarrow T_e'}$. Common examples of such systems are (causal) dynamical and hybrid dynamical systems [7,34].

Conformance testing Let $\mathfrak{A}_1$ and $\mathfrak{A}_2$ be systems and let $\mathcal{D}_{\mathcal{TR}}$ be a metric over output traces. For a set Π_{ip} of input traces, we define the *(quantitative) conformance* between $\mathfrak{A}_1$ and $\mathfrak{A}_2$ w.r.t. Π_{ip} as $\sup_{\pi_{\mathsf{ip}} \in \Pi_{\mathsf{ip}}} \mathcal{D}_{\mathcal{TR}}\left(\mathfrak{A}_1\left(\pi_{\mathsf{ip}}\right), \mathfrak{A}_2\left(\pi_{\mathsf{ip}}\right)\right)$ The conformance between $\mathfrak{A}_1$ and $\mathfrak{A}_2$ is their conformance w.r.t. the set of all input traces.

The conformance testing problem asks, given systems $\mathfrak{A}_1, \mathfrak{A}_2$, a trace metric $\mathcal{D}_{\mathcal{TR}}$, a tolerance δ, and a set of test input traces Π_{test}, if the quantitative conformance between $\mathfrak{A}_1$ and $\mathfrak{A}_2$ w.r.t. Π_{test} is at most δ. Clearly, conformance w.r.t. Π_{test} is a lower bound on the conformance between $\mathfrak{A}_1$ and $\mathfrak{A}_2$.

Algorithm 1 is a standard optimization-guided adaptive testing algorithm. To define the set Π_{test} of test inputs, we use a fixed finite parameterization of the input space using a finite set F of *basis functions* and fix a time horizon T. We only generate inputs obtained as a linear combination $\sum_{f \in F} p_f \cdot f$ of basis functions over the interval $[0, T]$, where the coefficients $\{p_f \mid f \in F\}$ come from a closed convex subset of $\mathbb{R}^{|F|}$.

In each step, Algorithm 1 picks an input signal u and computes the distance between the corresponding outputs $y_1 = \mathfrak{A}_1(u)$ and $y_2 = \mathfrak{A}_2(u)$. Based on heuristics that rely on the

Algorithm 1: Algorithm for Conformance Testing

Input: Systems $\mathfrak{A}_1, \mathfrak{A}_2$, trace metric $\mathcal{D}_{\mathcal{TR}}$, time horizon T, input parameterization F, termination criterion `terminate?`

Output: Input u that achieves maximum distance between $\mathfrak{A}_1$ and $\mathfrak{A}_2$

```
d ← 0, u ←⊥, dmax ← 0, umax ←⊥
while not(terminate?) do
    u ← pickNewInputs(F, T, d)
    y1 ← simulate(A1, u, T) and y2 ← simulate(A2, u, T)
    d ← D_TR(y1, y2)
    if d > dmax then dmax ← d, umax ← u
end
return "on input umax, outputs A1(umax) and A2(umax) differ by dmax by time T"
```

current distance, and a possibly bounded history of costs, the procedure then picks a new value for u by choosing new values for the coefficients $\{p_f \mid f \in F\}$. For instance, in a gradient-ascent based procedure, the new value of u is chosen by estimating the local gradient in each direction in the input-parameter space, and then picking the direction that has the largest (positive) gradient. In our implementation, we use the Nelder-Mead (or nonlinear simplex) algorithm to pick new inputs.

On termination (e.g., when some maximum number of iterations is reached), the algorithm returns the conformance distance between $\mathfrak{A}_1$ and $\mathfrak{A}_2$ w.r.t. the set of tests generated. One can compare the distance to some tolerance δ chosen based on engineering requirements.

Sampling schemes and resulting interpolated traces In practice, the output behaviors of the systems are observed with a sampling process, thus y_1 and y_2 on line 4 are discrete time-sampled *sequences*. We go from these sequences to output traces either by linear interpolation between the sampled time points, or by assuming a constant value in between the sampled values in a sample-and-hold scheme.

In the first case, we get a polygonal (piecewise linear) trace. Formally, a *polygonal trace* $\pi : I_\pi \to \mathcal{O}$ where $\mathcal{O}$ is a vector space with the scalar field $\mathbb{R}$ is a continuous trace such that there exists a finite sequence $\min I_\pi = t_0 < t_1 < \cdots < t_m = \max I_\pi$ of time-points such that the trace segment between t_k and t_{k+1} is affine for all $0 \leq k < m$, i.e., for $t_k \leq t \leq t_{k+1}$ we have $\pi(t) = \pi(t_k) + \frac{t-t_k}{t_{k+1}-t_k} \cdot (\pi(t_{k+1}) - \pi(t_k))$.

In the sample-and-hold case, we get a *piecewise constant* trace $\pi : I_\pi \to \mathcal{O}$ for which there exists a finite sequence $\min I_\pi = t_0 < t_1 < \cdots < t_m = \max I_\pi$ of time-points such that π is constant over the left-closed right-open intervals $[t_i, t_{i+1})$ for $0 \leq i < m$, and constant over $[t_{m-1},\ \max I_\pi]$ (the last interval is also right closed). A piecewise constant trace is right continuous, but need not be left continuous.

Given a timed trace sequence tseq, let $[\![\mathsf{tseq}]\!]_\Xi$ denote the polygonal or piecewise constant trace obtained from tseq by the linear interpolation or sample-and-hold scheme Ξ. Let $\mathsf{tseq}_\pi, \mathsf{tseq}_{\pi'}$ be two corresponding samplings of the traces π, π', respectively. For a trace metric $\mathcal{D}_{\mathcal{TR}}$, we have:

$$\begin{aligned}\mathcal{D}_{\mathcal{TR}}(\pi, \pi') \leq\ & \mathcal{D}_{\mathcal{TR}}\left([\![\mathsf{tseq}_\pi]\!]_\Xi, [\![\mathsf{tseq}_{\pi'}]\!]_\Xi\right) + \mathcal{D}_{\mathcal{TR}}\left([\![\mathsf{tseq}_\pi]\!]_\Xi, \pi\right) \\ & + \mathcal{D}_{\mathcal{TR}}\left([\![\mathsf{tseq}_{\pi'}]\!]_\Xi, \pi'\right).\end{aligned}$$

If $\Delta_\Xi^{\mathsf{samerr}}$ is a bound on the distance between a trace and an interpolated completion according to scheme Ξ of the sampling, we have that $\mathcal{D}_{\mathcal{TR}}(\pi, \pi') \leq \mathcal{D}_{\mathcal{TR}}([\![\mathsf{tseq}_\pi]\!]_\Xi, [\![\mathsf{tseq}_{\pi'}]\!]_\Xi) + 2 \cdot \Delta_\Xi^{\mathsf{samerr}}$. Thus, a value of $2 \cdot \Delta_\Xi^{\mathsf{samerr}}$ needs to be added in the testing algorithm to account for the error due to the polygonal or sample-and-hold approximations.

2.2 The Skorokhod metric

We now define the Skorokhod metric, which we use as the metric in Algorithm 1.

A *retiming* $\mathsf{r} : I \to I'$, for closed intervals I, I' of $\mathbb{R}_+$ is an order-preserving (i.e., monotone strictly-increasing) continuous bijective function from I to I'; thus if $t < t'$ then $\mathsf{r}(t) < \mathsf{r}(t')$. Let $\mathsf{R}_{I \to I'}$ be the class of retiming functions from I to I' and let $\mathcal{I}$ be the identity retiming. Intuitively, retiming can be thought of as follows: imagine a stretchable and compressible timeline; a retiming of the original timeline gives a new timeline where some parts have been stretched, and some compressed, without the timeline having been broken. Given a trace $\pi : I_\pi \to \mathcal{O}$, and a retiming $\mathsf{r} : I \to I_\pi$; the function $\pi \circ \mathsf{r}$ is another trace from I to $\mathcal{O}$.

Definition 1 (*Skorokhod metric*) Given a retiming $\mathsf{r} : I \to I'$, let $||\,\mathsf{r} - \mathcal{I}\,||_{\sup}$ be defined as $||\,\mathsf{r} - \mathcal{I}\,||_{\sup} = \sup_{t \in I} |\,\mathsf{r}(t) - t|$. Given two traces $\pi : I_\pi \to \mathcal{O}$ and $\pi' : I_{\pi'} \to \mathcal{O}$, where $\mathcal{O}$ is a metric space with the associated metric $\mathcal{D}_{\mathcal{O}}$, and a retiming $\mathsf{r} : I_\pi \to I_{\pi'}$, let $\left\|\pi \ - \ \pi' \circ \mathsf{r}\right\|_{\sup}$ be defined as:

$$\left\|\pi \ - \ \pi' \circ \mathsf{r}\right\|_{\sup} = \sup_{t \in I_\pi} \mathcal{D}_{\mathcal{O}}\left(\pi(t)\ ,\ \pi'\left(\mathsf{r}(t)\right)\right).$$

The *Skorokhod distance*[4] between the traces $\pi()$ and $\pi'()$ is defined to be:

$$\mathcal{D}_{\mathcal{S}}(\pi, \pi') = \inf_{r \in \mathsf{R}_{I_\pi \to I_{\pi'}}} \max\left(\|\mathsf{r} - \mathcal{I}\|_{\sup}\ ,\ \left\|\pi \ - \ \pi' \circ \mathsf{r}\right\|_{\sup}\right). \tag{1}$$

□

Intuitively, the Skorokhod distance incorporates two components: the first component quantifies the *timing discrepancy* of the timing distortion required to "match" two traces, and the second quantifies the *value mismatch* (in the metric space $\mathcal{O}$) of the values under the timing distortion. In the retimed trace $\pi \circ \mathsf{r}$, we see exactly the same values as in π, in exactly the same order, but the times at which the values are seen can be different.

2.3 Skorokhod metric computation: piecewise constant traces

In this subsection we derive a procedure for computing the Skorokhod distance between piecewise constant traces. The outline of the subsection is as follows. First in Lemma 1, we show that we can use non-decreasing functions as retimings (instead of allowing only strictly increasing functions). Then, in Lemmas 2 and 3, we show that for piecewise constant traces, the values of $\sup_{t \in I_\pi} \mathcal{D}_{\mathcal{O}}\left(\pi(t)\ ,\ \pi'\left(\mathsf{r}(t)\right)\right)$ and $\sup_{t \in I} |\,\mathsf{r}(t) - t|$ used in the definition of the Skorokhod metric (Definition 1) can be determined by looking at a *finite* set of timepoints. Using the this result, we then obtain a class of ϵ-optimal retimings in Lemma 4. Finally, using Lemmas 3 and 4, we devise a dynamic programming algorithm to compute the Skorokhod distance between piecewise constant traces. The main result of this subsection is stated in Theorem 1. The reader may skip directly to Theorem 1 without hampering readability of the rest of the paper.

We assume for simplicity of presentation that both π, π' contain m segments, are over the same intervals, and moreover that both traces are the result of sampling at the same time instants (the inter-sample time duration may be variable). As a first step, we prove the retimings can be *non-decreasing* and onto, rather than monotone strictly-increasing and bijective

Lemma 1 (Non-decreasing retimings for piecewise constant traces) *The value in Eq.* (1) *is unchanged for piecewise constant traces if* non-decreasing *and onto retimings are allowed. That is, for π, π' piecewise constant traces, we have:*

$$\mathcal{D}_{\mathcal{S}}(\pi, \pi') = \inf_{\substack{r:\, I_\pi \to I_{\pi'} \\ \mathsf{r}\ \textit{non-decreasing and onto}}} \max\left(\|\mathsf{r} - \mathcal{I}\|_{\sup}\ ,\ \left\|\pi \ - \ \pi' \circ \mathsf{r}\right\|_{\sup}\right). \tag{2}$$

Proof Let π, π' be over the time-intervals I_π $(= I_{\pi'})$. Suppose I_π consists of disjoint intervals $J_0, I_0, J_1, I_1, \ldots, I_a, J_a$ (in order) such that r is constant over the I intervals, and strictly increasing over the J intervals (J_0 or J_a may be empty, but other J intervals are non-empty).

[4] The two components of the Skorokhod distance (the retiming, and the value difference components) can be weighed with different weights—this simply corresponds to a change of scale.

Fix $\epsilon > 0$. Consider I_0. Since I_π and $I_{\pi'}$ contain more than one time-point, and and r is onto, at least one of J_0, J_1 is non-empty. We can show that we can "wiggle" the retiming r to get another retiming r_0 such that

1. r_0 is monotone increasing over J_0, I_0, J_1 and J_k for $k \geq 2$.
2. r_0 is equal to r over $I_1, J_2, I_2, \ldots, J_a$.
3. $\|\mathsf{r}_0 - \mathsf{r}\|_{\sup} < \epsilon$ (over J_0, I_0, J_1).
4. $\pi' \circ \mathsf{r}(t) = \pi' \circ \mathsf{r}_0(t)$ for all $t \in I_\pi$, which implies that $\left\|\pi - \pi' \circ \mathsf{r}\right\|_{\sup} = \left\|\pi - \pi' \circ \mathsf{r}_0\right\|_{\sup}$.

That is, we locally perturb r a little bit so that it becomes monotone increasing over I_0, and the perturbation does not affect the trace matchings between π and π' under the retimings.

Repeating the procedure, we get r_e such that

1. $\mathsf{r}_e : I_\pi \to I_{\pi'}$ is monotone strictly increasing and bijective.
2. $\|\mathsf{r}_e - \mathsf{r}\|_{\sup} < \epsilon$.
3. $\pi' \circ \mathsf{r}(t) = \pi' \circ \mathsf{r}_e(t)$ for all $t \in I_\pi$, which implies that $\left\|\pi - \pi' \circ \mathsf{r}\right\|_{\sup} = \left\|\pi - \pi' \circ \mathsf{r}_e\right\|_{\sup}$.

Thus, for every $\epsilon > 0$, given a non-decreasing and onto retiming r, there exists a strictly increasing and bijective retiming r_e such that

$$\max\left(\|\mathsf{r}_e - \mathsf{r}\|_{\sup}, \left\|\pi' \circ \mathsf{r}(t) - \pi' \circ \mathsf{r}_e(t)\right\|_{\sup}\right) < \epsilon$$

This implies that the value of Eq. (1) does not change if we allow non-decreasing retimings. This complete the proof of the lemma (the details on how to perform the perturbations can be found in the Appendix). □

We now show that when computing the value of Eq. (2), given a retiming r, we only need to look at the values of $\mathsf{r}(t) - \mathcal{I}(t)$ and $\pi(t) - \pi' \circ \mathsf{r}(t)$ at a finite set of timepoints $t \in \{t_0, t_1, \ldots, t_m\}$. For this, we need the following lemma.

Lemma 2 (Retiming function specification for the purpose of Skorokhod distance computation) *Given π, π' piecewise constant traces such that π and π' are constant over the disjoint intervals $[t_0, t_1), [t_1, t_2), \ldots, [t_{m-1}, t_m]$ (with $I_\pi = I_{\pi'} = [t_0, t_m]$, and a non-decreasing and onto retiming function $\mathsf{r} : I_\pi \to I_{\pi'}$, consider*

$$\left\|\pi - \pi' \circ \mathsf{r}\right\|_{\sup} \tag{3}$$

The value of Expression (3) is dependent only on the value of r at timepoints $t_0, t_1, \ldots, t_m$, that it, if r_1 and r_2 coincide on $t_0, t_1, \ldots, t_m$, then the value of Expression (3) is the same whether r_1 or r_2 is used as the retiming function in the expression.

Proof Assume $m > 1$ (otherwise both traces have one segment and the claim is easy to prove. Since π, π' are piecewise constant, and r is non-decreasing, we have

$$\begin{aligned}&\left\{\pi(t) - \pi' \circ \mathsf{r}(t) \mid t \in I_\pi\right\} \\ &\quad = \cup_{0 \leq k < m} \left\{\pi(t_k) - \pi'(t') \mid t' \in \left[\mathsf{r}(t_k), \mathsf{r}(t_{k+1})\right) \cap \{t_0, t_1, \ldots, t_m\}\right\}\end{aligned}$$

The above expression can be understood as follows: r maps the time interval $[t_k, t_{k+1})$ to the time interval $[\mathsf{r}(t_k), \mathsf{r}(t_{k+1}))$. Thus, the π segment value $\pi(t_k)$ is matched to the values $\left\{\pi'(t') \mid t' = [\mathsf{r}(t_k), \mathsf{r}(t_{k+1})) \cap \{t_0, t_1, \ldots, t_m\}\right\}$; using the fact that π' only changes values at $\{t_0, t_1, \ldots, t_m\}$. The claim follows. □

Consider the expression

$$\max\left(\|\mathsf{r}-\mathcal{I}\|_{\sup},\ \left\|\pi\ -\ \pi'\circ\mathsf{r}\right\|_{\sup}\right).$$

which expands to

$$\max\left(\sup_{t\in I_\pi}|\mathsf{r}(t)-\mathcal{I}(t)|,\ \sup_{t\in I_\pi}\mathcal{D}_{\mathcal{O}}\left(\pi(t),\ \pi'\circ\mathsf{r}(t)\right)\right). \tag{4}$$

Lemma 2, and its proof hint that instead of taking the max over the whole interval I_π in Eq. (4), we can take the max over the finite set of timepoints $\{t_0, t_1, \ldots, t_m\}$.

Technically, it turns out this claim is not true because of the $\sup_{t\in I_\pi}|\mathsf{r}(t)-\mathcal{I}(t)|$ part: the retiming function r may deviate more from $\mathcal{I}$ on $I_\pi \setminus \{t_0, t_1, \ldots, t_m\}$ than on $\{t_0, t_1, \ldots, t_m\}$. However, given a retiming function r, we can always find another retiming function $\mathsf{r}^\dagger$ such that (i) $\mathsf{r}^\dagger$ agrees with r over $\{t_0, t_1, \ldots, t_m\}$; (ii) is closer to $\mathcal{I}$; and (iii) $\sup_{t\in I_\pi}\left|\mathsf{r}^\dagger(t)-\mathcal{I}(t)\right|$ can be determined by the values of $\mathsf{r}^\dagger$ on $\{t_0, t_1, \ldots, t_m\}$. That is,

1. $\mathsf{r}^\dagger(t)=\mathsf{r}(t)$ for $t\in\{t_0, t_1, \ldots, t_m\}$, and
2. $\sup_{t\in I_\pi}|\mathsf{r}(t)-\mathcal{I}(t)| \geq \sup_{t\in I_\pi}\left|\mathsf{r}^\dagger(t)-\mathcal{I}(t)\right|$, and
3. $\sup_{t\in I_\pi}\left|\mathsf{r}^\dagger(t)-\mathcal{I}(t)\right| = \max\sup_{t\in\{t_0,t_1,\ldots,t_m\}}\left|\mathsf{r}^\dagger(t)-\mathcal{I}(t)\right|$.

This together with the proof of Lemma 2 gives us the following result.

Lemma 3 (Sufficiency of giving retiming function values only on a finite set) *Given π, π' piecewise constant traces such that π and π' are constant over the disjoint intervals $[t_0, t_1), [t_1, t_2), \ldots, [t_{m-1}, t_m]$ (with $I_\pi = I_{\pi'} = [t_0, t_m]$), we have*

$$\mathcal{D}_{\mathcal{S}}(\pi,\pi') = \inf_{\substack{r:\, I_\pi\to I_{\pi'}\\ \mathsf{r}\ \textit{non-decreasing}\\ \textit{and onto}}}\max\left(\begin{array}{c}\max_{t\in\{t_0,t_1,\ldots,t_m\}}|\mathsf{r}(t)-\mathcal{I}(t)|,\\ \max\limits_{0\leq k<m}\ \max\limits_{t'\in\left[\mathsf{r}(t_k),\mathsf{r}(t_{k+1})\right)\cap\{t_0,t_1,\ldots,t_m\}}\mathcal{D}_{\mathcal{O}}\left(\pi(t_k),\ \pi'(t')\right)\end{array}\right). \tag{5}$$

□

Note that in Eq. (5), the quantity

$$\max_{t'\in\left[\mathsf{r}(t_k),\mathsf{r}(t_{k+1})\right)\cap\{t_0,t_1,\ldots,t_m\}}\mathcal{D}_{\mathcal{O}}\left(\pi(t_k),\ \pi'(t')\right)$$

for a given t_k gives the maximal discrepancy arising as a result of $\pi(t_k)$ being matched to the π' values $\left\{\pi'(t')\mid t'=[\mathsf{r}(t_k),\mathsf{r}(t_{k+1}))\cap\{t_0,t_1,\ldots,t_m\}\right\}$.

ϵ-optimal retiming functions Finally, we construct ϵ-optimal retiming functions in the minimization of Eq. (5) (in general, optimal retiming functions need not exist as Eq. (5) still has an "inf" over retimings). Fix ϵ such that $\min_{0\leq k<m}(t_{k+1}-t_k) > \epsilon > 0$. Consider a class C_ϵ of non-decreasing retiming functions such that for $\mathsf{r}\in C_\epsilon$, for all k, there is some j such that $\mathsf{r}(t_k)=t_j$, or $t_j-\epsilon$. That it, the retiming r maps the trace change timepoints t_k to other change time-points, or to just before other change timepoints. The following lemma shows that we can restrict retimings to belong to this class (the proof can be found in the appendix).

Lemma 4 (ϵ-optimal retiming functions) *Let π, π' be piecewise constant traces, and let C_ϵ be the class of non-decreasing retiming functions as defined above. Consider*

$$\mathcal{D}_{\mathcal{S}}^{C_\epsilon}(\pi,\pi') = \inf_{r\in C_\epsilon}\max\left(\begin{array}{c}\max_{t\in\{t_0,t_1,\ldots,t_m\}}|\mathsf{r}(t)-\mathcal{I}(t)|,\\ \max\limits_{0\leq k<m}\ \max\limits_{t'\in\left[\mathsf{r}(t_k),\mathsf{r}(t_{k+1})\right)\cap\{t_0,t_1,\ldots,t_m\}}\mathcal{D}_{\mathcal{O}}\left(\pi(t_k),\ \pi'(t')\right)\end{array}\right) \tag{6}$$

We have

$$\mathcal{D}_{\mathcal{S}}^{C_\epsilon}(\pi, \pi') \;\leq\; \mathcal{D}_{\mathcal{S}}(\pi, \pi') + \epsilon.$$

□

Lemma 4 shows that we can search over a finite space of retiming functions to compute $\mathcal{D}_{\mathcal{S}}(\pi, \pi')$. The range of the retiming functions, over the set $\{t_0, t_1, \ldots, t_m\}$ can be restricted to $\{t_0, t_1^-, t_1, t_2^-, t_2, \ldots, t_{m-1}^-, t_{m-1}, t_m\}$, where t_k^- is the timepoint arbitrarily close to (and less than) t_k. Using this fact, we design a dynamic programming algorithm to compute $\mathcal{D}_{\mathcal{S}}(\pi, \pi')$ below.

Observe that we "push" the discrepancy $\max_{t\in\{t_0,t_1,\ldots,t_m\}} |\mathsf{r}(t)|$ inside the second term as follows.

$$\begin{aligned}
&\max\left(\begin{array}{c}\max_{t\in\{t_0,t_1,\ldots,t_m\}} |\mathsf{r}(t) - \mathcal{I}(t)|\,, \\ \max\limits_{0\leq k<m} \max\limits_{t'\in\left[\mathsf{r}(t_k),\mathsf{r}(t_{k+1})\right)\cap\{t_0,t_1,\ldots,t_m\}} \mathcal{D}_{\mathcal{O}}\left(\pi(t_k),\ \pi'(t')\right)\end{array}\right) \\
&\quad = \max_{0\leq k<m} \max_{t'\in\left[\mathsf{r}(t_k),\mathsf{r}(t_{k+1})\right)\cap\{t_0,t_1,\ldots,t_m\}} \max\left(\left|t_k - t'\right|,\ \mathcal{D}_{\mathcal{O}}\left(\pi(t_k),\ \pi'(t')\right)\right) \qquad (7)
\end{aligned}$$

For every a trace $\eta : I_\eta \to \mathcal{O}$, associate a *time-explicit* trace $\hat{\eta} : I_\eta \to \mathbb{R}_+ \times \mathcal{O}$ defined by $\hat{\eta}(t) = (t, \eta(t))$, and let $\mathcal{D}_{\hat{\mathcal{O}}}\Big((t_1, x_1),\ (t_2, x_2)\Big) = \max\Big(\,|t_1 - t_2|\,,\ \mathcal{D}_{\mathcal{O}}(x_1, x_2)\Big)$. It can be checked that Eq. (7) can be expressed as

$$\max_{0\leq k<m} \max_{t'\in\left[\mathsf{r}(t_k),\mathsf{r}(t_{k+1})\right)\cap\{t_0,t_1,\ldots,t_m\}} \mathcal{D}_{\hat{\mathcal{O}}}\left(\hat{\pi}(t),\ \hat{\pi}' \circ \mathsf{r}(t)\right)$$

And thus, Eq. (6) as:

$$\mathcal{D}_{\mathcal{S}}^{C_\epsilon}(\pi, \pi') = \inf_{r\in C_\epsilon} \max_{0\leq k<m} \max_{t'\in\left[\mathsf{r}(t_k),\mathsf{r}(t_{k+1})\right)\cap\{t_0,t_1,\ldots,t_m\}} \mathcal{D}_{\hat{\mathcal{O}}}\left(\hat{\pi}(t),\ \hat{\pi}' \circ \mathsf{r}(t)\right) \qquad (8)$$

Since the retimings can be restricted to $\{t_0, t_1^-, t_1, t_2^-, t_2, \ldots, t_{m-1}^-, t_{m-1}, t_m\}$ (as mentioned previously), we solve the above problem (8) as follows. In order to simplify notation, let $x_0, x_1, x_2, \ldots, x_{2m}$ be defined as

$$(t_0, \pi(t_0)),\ (t_1^-, \pi(t_1^-)),\ (t_1, \pi(t_1)), \ldots (t_{m-1}^-, \pi(t_{m-1}^-)),\ (t_{m-1}, \pi(t_{m-1})),\ (t_m, \pi(t_m))$$

where $\pi(t_k^-) = \pi(t_{k-1})$, and t_k^- is a time which is just less than t_k; that is $\pi(t_k^-)$ is the value of trace π just before time t_k. Let $y_0, y_1, y_2, \ldots, y_{2m}$ be defined similarly for π'.

We build a dynamic programming algorithm. Let $M(i, j)$ for $j > i$ denote the fact that point y_j is mapped to x_i (and possibly earlier x points); $M(i, j)$ for $i > j$ denote the fact that point y_j is mapped to x_i (and possibly later x points); and $M(i, i)$ denote the case when point y_i is mapped to x_i. For example, $M(2, 2m)$ denotes the fact that point y_{2m} has been moved "left" via retiming to match x_2 (and perhaps an earlier point). Note that this retiming has the effect of "stretching" y_{2m} – the single point y_{2m} now matches the x segment $[x_2, x_{2m}]$. The recurrence relation for M is set up as follows.

$$M(i, j) = \min\left(\begin{array}{l}\max\left(M(i-1, j-1),\ \mathcal{D}_{\hat{\mathcal{O}}}\left(x_i, y_j\right)\right) \\ \max\left(M(i, j-1),\ \mathcal{D}_{\hat{\mathcal{O}}}\left(x_i, y_j\right)\right) \\ \max\left(M(i-1, j),\ \mathcal{D}_{\hat{\mathcal{O}}}\left(x_i, y_j\right)\right)\end{array}\right) \qquad (9)$$

(in case $i - 1$ or $j - 1$ is negative, the min omits those lines). The base condition is given by $M(0, 0) = \mathcal{D}_{\hat{\mathcal{O}}}(x_0, y_0)$. The correctness of the recurrence relation follows from the fact that either

- The portion $[y_0,\ y_{j-1}]$ is mapped to $[x_0, x_{i-1}]$, and the portion $(y_{j-1},\ y_j]$ mapped to $(x_{i-1},\ x_i]$; or
- The portion $[y_0,\ y_{j-1}]$ is mapped to $[x_0, x_i]$, and the portion $(y_{j-1},\ y_j]$ mapped to the single point x_i; or
- The portion $[y_0,\ y_j]$ is mapped to $[x_0, x_{i-1}]$, and the portion $(x_{i-1},\ x_i]$ mapped to the single point y_j.

The recurrence relation (9) mirrors these three cases. Note that when we expand $M(i-1, j-1)$, we see the distance comparison between x_{i-1}, y_{j-1} (and similarly for the other cases). Since these are the only cases that can arise from non-decreasing retimings, we get that $M(2m, 2m)$ computes $\mathcal{D}_S(\pi, \pi')$. A similar algorithm with more bookkeeping applies when π, π' sample points are at different time instants. Thus, we get the following theorem.

Theorem 1 (Computing the Distance between Piecewise Constant Traces) *Let $\pi : I_\pi \to \mathbb{R}^n$ and $\pi' : I_{\pi'} \to \mathbb{R}^n$ be two piecewise constant traces with m_π and $m_{\pi'}$ affine segments respectively. Let the Skorokhod distance between them (for the $L \in \{L_1, L_2, L_\infty\}$ norm on $\mathbb{R}^n$) be denoted as $\mathcal{D}_S^L(\pi, \pi')$.*

1. *We can compute $\mathcal{D}_S^L(\pi, \pi')$ in time $O(m_\pi \cdot m_{\pi'} \cdot n)$.*
2. *Suppose we restrict retimings to be such that the i-th constant segment of π can only be matched to π' constant segments $i - W$ through $i + W$ for all i, where $W \geq 1$. Under this retiming restriction, we can compute $\mathcal{D}_S^L(\pi, \pi')$ with a streaming algorithm in time $O((m_\pi + m_{\pi'}) \cdot n \cdot W)$.* □

2.4 Skorokhod metric computation: polygonal traces

We devised an algorithm to compute the Skorokhod distance between polygonal traces in an earlier work [26]. Note after retiming, the retimed version $\pi \circ \mathsf{r}$ of a polygonal trace π need not be polygonal (see [26]), in spite of this, the algorithm is polynomial time.

Theorem 2 (Distance Monitoring between Polygonal Traces [27]) *Let $\pi : I_\pi \to \mathbb{R}^n$ and $\pi' : I_{\pi'} \to \mathbb{R}^n$ be two polygonal traces with m_π and $m_{\pi'}$ affine segments respectively. Let the Skorokhod distance between them (for the L_2 norm on $\mathbb{R}^n$) be denoted as $\mathcal{D}_S(\pi, \pi')$.*

1. *Given $\delta \geq 0$, it can be checked whether $\mathcal{D}_S(\pi, \pi') \leq \delta$ in time $O(m_\pi \cdot m_{\pi'} \cdot n)$.*
2. *Suppose we restrict retimings to be such that the i-th affine segment of π can only be matched to π' affine segments $i - W$ through $i + W$ for all i, where $W \geq 1$. Under this retiming restriction, we can determine, with a streaming algorithm, whether $\mathcal{D}_S(\pi, \pi') \leq \delta$ in time $O((m_\pi + m_{\pi'}) \cdot n \cdot W)$.* □

Let us denote by $\mathcal{D}_S^W(\pi, \pi')$ the Skorokhod difference between π, π' under the retiming restriction of the second part of Theorem 2, i.e., the value obtained by restricting the retimings in Eq. (1).[5] The value $\mathcal{D}_S^W(\pi, \pi')$ is an upper bound on $\mathcal{D}_S(\pi, \pi')$. In addition, for $W' < W$, we have $\mathcal{D}_S^W(\pi, \pi') \leq \mathcal{D}_S^{W'}(\pi, \pi')$. These results on the distances computed under the window restriction W also apply to piecewise constant traces (Theorem 1).

Computing the distance value between polygonal traces A polynomial time algorithm was presented in [27] to *compute* the distance value using the distance monitoring routine. However, even though polynomial time, and even under a sliding-window restriction, the time complexity of the algorithm is unsatisfactory for use in conformance testing—a careful analysis shows that that the time complexity of the algorithm for determining the distance value

[5] $\mathcal{D}_S^W$ is not a metric over traces (the triangle inequality fails).

using the procedure of [27] for a window size W is $O\left((m_\pi + m_{\pi'}) \cdot n \cdot W^2 \cdot \log(W)\right)$. In practice, it turned out to be more efficient to do a binary search employing the monitoring routine to obtain the distance value; we observed around 7 binary search calls on average (the typical window size was 100).

The range of $\mathcal{D}_{\mathrm{S}}^{W}(\pi, \pi')$ can be bound as follows for the binary search. Observe that $\pi(\min I_\pi)$ must be mapped to $\pi'(\min I_{\pi'})$ by any retiming (since retimings are onto), and similarly $\pi(\max I_\pi)$ must be mapped to $\pi'(\max I_{\pi'})$. Thus, a lower bound on $\mathcal{D}_{\mathrm{S}}^{W}(\pi, \pi')$ is:

$$\max\left(\mathcal{D}_{\mathcal{O}}\left(\pi(\min I_\pi),\ \pi'(\min I_{\pi'})\right),\ \mathcal{D}_{\mathcal{O}}\left(\pi(\max I_\pi),\ \pi'(\max I_{\pi'})\right)\right).$$

The most obvious upper bound is to fix a retiming r (for example the identity function in case the time-domains of both traces match), and compute $\max\left(\|\mathsf{r} - \mathcal{I}\|_{\sup}, \left\|\pi - \pi' \circ \mathsf{r}\right\|_{\sup}\right)$. We can also use the piecewise constant procedure given in the last subsection to get better bounds. Given a polygonal trace ξ, let ξ_{pwc} denote the corresponding piecewise constant trace obtained by sampling ξ at the endpoints of the affine segments of ξ (and using sample-and-hold in between the sample points). Suppose traces π, π' be over the time interval $[t_0, t_m]$, and let $t_0 < t_1 < \ldots t_m$ be such that both traces are affine between t_k and t_{k+1} for all k. It can be checked that the pointwise distance between π, and the corresponding piecewise constant trace π_{pwc}, defined as

$$\mathcal{D}_{\sup}\left(\pi, \pi_{\mathsf{pwc}}\right) \;=\; \sup_{t \in I_\pi} \mathcal{D}_{\mathcal{O}}\left(\pi(t), \pi_{\mathsf{pwc}}(t)\right) \tag{10}$$

is equal to

$$\max\left(\mathcal{D}_{\mathcal{O}}\left(\pi(t_1), \pi_{\mathsf{pwc}}(t_0)\right),\ \mathcal{D}_{\mathcal{O}}\left(\pi(t_2), \pi_{\mathsf{pwc}}(t_1)\right), \ldots, \mathcal{D}_{\mathcal{O}}\left(\pi(t_m), \pi_{\mathsf{pwc}}(t_{m-1})\right)\right).$$

We have

$$\mathcal{D}_{\mathrm{S}}^{W}\left(\pi, \pi'\right) \;\leq\; \mathcal{D}_{\mathrm{S}}^{W}\left(\pi_{\mathsf{pwc}}, \pi'_{\mathsf{pwc}}\right) \;+\; \mathcal{D}_{\sup}\left(\pi, \pi_{\mathsf{pwc}}\right) \;+\; \mathcal{D}_{\sup}\left(\pi', \pi'_{\mathsf{pwc}}\right).$$

The above inequality follows from the triangle inequality over $\mathcal{D}_{\mathcal{O}}$. Similarly,

$$\mathcal{D}_{\mathrm{S}}^{W}\left(\pi_{\mathsf{pwc}}, \pi'_{\mathsf{pwc}}\right) \;\leq\; \mathcal{D}_{\mathrm{S}}^{W}\left(\pi, \pi'\right) \;+\; \mathcal{D}_{\sup}\left(\pi, \pi_{\mathsf{pwc}}\right) \;+\; \mathcal{D}_{\sup}\left(\pi', \pi'_{\mathsf{pwc}}\right).$$

This gives us the following bounds on $\mathcal{D}_{\mathrm{S}}^{W}(\pi, \pi')$

Proposition 1 *Let π, π' be polygonal traces over the time interval $[t_0, t_m]$ (with values in the vector space $\mathcal{O}$). The Skorokhod distance $\mathcal{D}_{\mathrm{S}}^{W}(\pi, \pi')$ under the window W retiming restriction lies in the interval $[\alpha_{\min}, \alpha_{\max}]$, where*

$$\alpha_{\min} = \max\left(\begin{array}{l}\mathcal{D}_{\mathcal{O}}\left(\pi(t_0),\ \pi'(t_0)\right),\ \mathcal{D}_{\mathcal{O}}\left(\pi(t_m),\ \pi'(t_m)\right), \\ \mathcal{D}_{\mathrm{S}}^{W}\left(\pi_{\mathsf{pwc}}, \pi'_{\mathsf{pwc}}\right) - \left(\mathcal{D}_{\sup}\left(\pi, \pi_{\mathsf{pwc}}\right) + \mathcal{D}_{\sup}\left(\pi', \pi'_{\mathsf{pwc}}\right)\right)\end{array}\right)$$

$$\alpha_{\max} = \min\left(\mathcal{D}_{\sup}\left(\pi, \pi'\right),\ \mathcal{D}_{\mathrm{S}}^{W}\left(\pi_{\mathsf{pwc}}, \pi'_{\mathsf{pwc}}\right) + \left(\mathcal{D}_{\sup}\left(\pi, \pi_{\mathsf{pwc}}\right) + \mathcal{D}_{\sup}\left(\pi', \pi'_{\mathsf{pwc}}\right)\right)\right)$$

with $\mathcal{D}_{\sup}$ for traces η, η' (over the same time interval I_η) defined as $\mathcal{D}_{\sup}(\eta, \eta') = \sup_{t \in I_\eta} \mathcal{D}_{\mathcal{O}}(\eta(t), \eta'(t))$. □

Note that $\mathcal{D}_{\sup}\left(\pi, \pi_{\mathsf{pwc}}\right)$ and (similarly for π') can be computed, for $\mathcal{O} = \mathbb{R}^n$ with norm L_1, L_2 or L_∞, in time $O(m \cdot n)$ (where m is the number of affine segments in π).

3 Transference of logical properties

In this section, we demonstrate a transference result involving the Skorokhod metric for the linear time logic FLTL (Freeze LTL) — a logic which augments LTL with *freeze* quantifiers [4] over both time and trace values. The logic we consider generalizes MTL, STL, TLTL [4], and STL* [8,9]. We show that if the Skorokhod distance between two traces is small, they satisfy close FLTL formulae. Given a formula ϕ of FLTL satisfied by trace π_1, we can compute a "relaxation" of ϕ that will be satisfied by the "close" trace π_2.

We first present the results in a propositional framework for the logic TLTL (obtained by augmenting LTL with freeze quantifiers over time), and then extend to $\mathbb{R}^n$-valued spaces for the logic FLTL which also has freeze quantifiers over $\mathbb{R}^n$-valued variables.

3.1 The logic TLTL

Let $\mathcal{P}$ be a set of propositions. A *propositional trace* π over $\mathcal{P}$ is a trace where the topological space is $2^{\mathcal{P}}$, with the associated metric $\mathcal{D}_{\mathcal{P}}(\sigma, \sigma') = 0$ if $\sigma = \sigma'$, and ∞ otherwise, for $\sigma, \sigma' \in 2^{\mathcal{P}}$. The set of all timed propositional traces over $\mathcal{P}$ is denoted by $\Pi(\mathcal{P})$. Note that if a trace has finite variability, i.e., if there exists a finite partition of $\mathsf{tdom}(\pi)$ into disjoint subintervals $I_0, I_1, \ldots, I_m$ such that π is constant on each subinterval, and in addition if $I_0, I_1, \ldots, I_{m-1}$ are left-closed and right open (with I_m closed on both sides), then the propositional trace π can be viewed as a trace obtained under a sample-and-hold scheme from a finite set of sample points.

Definition 2 (*TLTL*($\mathcal{F}_\mathsf{T}$) *Syntax*) Given a set of propositions $\mathcal{P}$, a set of (time) variables V_T, and a set $\mathcal{F}_\mathsf{T}$ of functions from $\mathbb{R}^l_+$ to $\mathbb{R}$, the formulae of TLTL($\mathcal{F}_\mathsf{T}$) are defined by the following grammar.

$$\phi := p \mid \text{TRUE} \mid f_\mathsf{T}(\overline{x}) \sim 0 \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \,\mathcal{U}\, \phi_2 \mid x.\phi$$

where

- $p \in \mathcal{P}$ and $x \in V_\mathsf{T}$, and $\overline{x} = (x_1, \ldots, x_l)$ with $x_i \in V_\mathsf{T}$ for all $1 \leq i \leq l$;
- $f_\mathsf{T} \in \mathcal{F}_\mathsf{T}$ is a real-valued function, and $\sim$ is one of $\{\leq, <, \geq, >\}$. □

The quantifier "x." is known as the *freeze quantifier*, and binds variable x to the current time. A variable x is defined to be *free* in ϕ as follows. The variable x is *not* free in $x.\Psi$, or in p (a proposition), or in TRUE, or in $f_\mathsf{T}(x_1, \ldots, x_l) \sim 0$ where $x_i \neq x$ for all i. It is also not free in ϕ if ϕ does not contain an occurrence of x. It is free in $\neg\psi$ iff x is free in ψ; and it is free in $\phi_1 \mathbin{\overset{\wedge}{\vee}} \phi_2$, or in $\phi_1 \,\mathcal{U}\, \phi_2$, iff x is free in either ϕ_1 or in ϕ_2. Finally, variable x is free in $f_\mathsf{T}(x_1, \ldots, x_l) \sim 0$ if some x_i is x. A formula is *closed* if it has no free variables.

Definition 3 (*TLTL*($\mathcal{F}_\mathsf{T}$) *Semantics*) Let $\pi : I \to 2^{\mathcal{P}}$ be a timed propositional trace, and let $\mathcal{E} : V_\mathsf{T} \mapsto I$ be the time environment mapping the variables in V_T to time values in I. The satisfaction of the trace π with respect to the TLTL($\mathcal{F}_\mathsf{T}$) formula ϕ in the time environment $\mathcal{E}$ is written as $\pi \models_\mathcal{E} \phi$, and is defined inductively as follows (denoting $t_0 = \min \mathsf{tdom}(\pi)$).

$\pi \models_{\mathcal{E}} p$ for $p \in \mathcal{P}$ iff $p \in \pi(t_0)$; $\quad \pi \models_{\mathcal{E}} \text{TRUE}$; $\quad \pi \models_{\mathcal{E}} \neg\Psi$ iff $\pi \not\models_{\mathcal{E}} \Psi$;

$\pi \models_{\mathcal{E}} \phi_1 \wedge \phi_2$ iff $\pi \models_{\mathcal{E}} \phi_1$ and $\pi \models_{\mathcal{E}} \phi_2$; $\quad \pi \models_{\mathcal{E}} \phi_1 \vee \phi_2$ iff $\pi \models_{\mathcal{E}} \phi_1$ or $\pi \models_{\mathcal{E}} \phi_2$;

$\pi \models_{\mathcal{E}} f_{\mathsf{T}}(x_1, \ldots, x_l) \sim 0$ iff $f_{\mathsf{T}}(\mathcal{E}(x_1), \ldots, \mathcal{E}(x_l)) \sim 0$ for $\sim \in \{\leq, <, \geq, >\}$;

$\pi \models_{\mathcal{E}} x.\psi$ iff $\pi \models_{\mathcal{E}[x:=t_0]} \psi$ where $\mathcal{E}[x := t_0]$ agrees with $\mathcal{E}$ for all $x_i \neq x$, and maps x to t_0;

$\pi \models_{\mathcal{E}} \phi_1 \mathcal{U} \phi_2$ iff $\pi^t \models_{\mathcal{E}} \phi_2$ for some $t \in I$ and $\pi^{t'} \models_{\mathcal{E}} \phi_1 \vee \phi_2$ for all $t_0 \leq t' < t$.

A timed trace π is said to satisfy the closed formula ϕ (written as $\pi \models \phi$) if there is some environment $\mathcal{E}$ such that $\pi \models_{\mathcal{E}} \phi$. □

We define additional temporal operators in the standard way: the "eventually" operator $\Diamond\phi$ stands for $\text{TRUE}\,\mathcal{U}\,\phi$; and the "always" operator $\Box\phi$ stands for $\neg\Diamond\neg\phi$. TLTL($\mathcal{F}_{\mathsf{T}}$) provides a richer framework than MTL [25] for expressing timing constraints as: (i) freeze quantifiers allow specification of constraints between distant contexts, which the bounded temporal operators in MTL cannot do; and (ii) the predicates $f_{\mathsf{T}}() \sim 0$ for $f_{\mathsf{T}} \in \mathcal{F}_{\mathsf{T}}$ allow the specification of complex timing requirements not expressible in MTL. Note that even if the predicates $f_{\mathsf{T}}() \sim 0$ are restricted to be of the form $x_1 - x_2 + c \sim 0$, where x_1, x_2 are freeze variables, and c is a constant, TLTL($\mathcal{F}_{\mathsf{T}}$) is more expressive than MTL [6] (and hence more expressive than MITL on which STL is based).

Example 1 (*Freeze quantification*) Suppose we want to express that whenever the event Q occurs, it is followed later by R, and then by S, such that the time difference between occurrences of Q and R is at most 5, and also the time difference between occurrences of Q and S is at most 10. This can be expressed in TLTL($\mathcal{F}_{\mathsf{T}}$) as

$$\Box\Big(x.Q \rightarrow \Diamond\big(y.\big[R \wedge (y \leq x + 5) \wedge \Diamond\,(z.\,(S \wedge z \leq x + 10))\,\big]\big)\Big).$$

Thus, freeze quantification, by giving a mechanism to bind times to variables, allows us to relate, with several constraints, far apart events. □

Example 2 (*Freeze quantification functions*) Suppose we want to express that whenever the event Q occurs, it must be followed by a response R within time λ^{t_Q} for some $\lambda > 1$ where t_Q is the time at which Q occurred; thus, the later Q occurs the more delay we can tolerate in the response time. The requirement can be expressed as $x.\,\big(Q \rightarrow \Diamond\big(y.\,(R \wedge (y - x \leq \lambda^x))\big)\big)$. □

Example 3 (*TLTL($\mathcal{F}_{\mathsf{T}}$) subsumes MTL*) Let $\mathcal{F}_{\mathsf{T}}$ be the set of two variable functions of the form $f(x, y) = x - y + c$ where c is a rational constant. Then TLTL($\mathcal{F}_{\mathsf{T}}$) subsumes MTL. The MTL formula $p\,\mathcal{U}_{[a,b]}q$ can be written as

$$x.\Big(p\,\mathcal{U}\,y.\big((y \leq x + b) \wedge (y \geq x + a) \wedge q\big)\Big).$$

We explain the formula as follows. We assign the "current" time t_x to the variable x, and some future time t_y to the variable y. The values t_x and t_y are such that at time t_y, we have q to be true, and moreover, at all times between t_x and t_y, we have $p \vee q$ to be true. Furthermore, t_y must be such that $t_y \in [t_x + a, t_x + b]$, which is specified by the term $(y \leq x + b) \wedge (y \geq x + a)$. □

Example 4 (*Temporal constraints*) Suppose we want to express that whenever the event p occurs, it must be followed by a response q, and then by r. In addition, we have the following

timing requirement: if $\varepsilon_{pq}, \varepsilon_{qr}, \varepsilon_{pr}$ are the time delays between p and q, between q and r, and between p and r, respectively, then: we must have $\varepsilon_{pq}^2 + \varepsilon_{qr}^2 + \varepsilon_{pr}^2 \leq d$ for a given positive constant d. This can be written using freeze quantifiers as the TLTL formula ϕ:

$$x.\big(p \rightarrow \Diamond\big(y.\big(q \wedge \Diamond\big[z.\big(r \wedge \big((y-x)^2 + (z-y)^2 + (z-x)^2 \leq d\big)\big]\big)\big)\big)\big).$$

□

3.2 Transference of TLTL properties for propositional traces

We now show that if a timed propositional trace π satisfies a TLTL($\mathcal{F}_\mathsf{T}$) formula ϕ, then any timed trace π' that is at most δ distance away from π satisfies a slightly relaxed version of the formula ϕ, the degree of relaxation being governed by δ; and the variance of the functions in $\mathcal{F}_\mathsf{T}$ over the time interval containing the time domains of π and π'.

We define the distance $\mathcal{D}_\mathcal{S}$ between two propositional traces as the Skorokhod distance, where we use $\mathcal{D}_\mathcal{P}$ as the distance between two sets of propositions.

Next, we define relaxations of TLTL($\mathcal{F}_\mathsf{T}$) formulae. The relaxations are defined as a syntactic transformation on formulae in negation-normal form, i.e., in which negations only appear at the propositions. It can be showed that every TLTL($\mathcal{F}_\mathsf{T}$) formula can be rewritten in negation-normal form, when we additionally use the waiting for operator, $\mathcal{W}$, defined as:

$$\begin{aligned}\pi \models_\mathcal{E} \phi_1 \,\mathcal{W}\, \phi_2 \text{ iff either (1) } \pi^t \models_\mathcal{E} \phi_1 \text{ for all } t \in I_\pi;\ \text{or (2) } \pi^t \models_\mathcal{E} \phi_2 \text{ for}\\ \text{some } t \in I_\pi; \textit{and} \pi^{t'} \models_\mathcal{E} \phi_1 \vee \phi_2 \text{ for all } \min I_\pi \leq t' < t.\end{aligned}$$

Removing negation using the $\mathcal{W}$ operator The following identities hold relating the $\mathcal{W}$ operator to the $\mathcal{U}$ operator

1. $\phi_1 \,\mathcal{U}\, \phi_2 \equiv \neg\,(\neg(\phi_2)\,\mathcal{W}\,(\neg\phi_1 \wedge \neg\phi_2))$; and
2. $\phi_1 \,\mathcal{W}\, \phi_2 \equiv \neg\,(\neg(\phi_2)\,\mathcal{U}\,(\neg\phi_1 \wedge \neg\phi_2))$.

Informally, the first identity states that $\neg(\phi_1 \,\mathcal{U}\, \phi_2)$ holds iff either (i) ϕ_2 never holds; or (ii) there is a point where ϕ_1 is false, and at that point and all points before it, ϕ_2 has remained false. The second identity is similar. The first identity above allows us to "push" the negations down using the $\mathcal{W}$ operator. The mechanism for the three interesting cases is below.

$$\begin{aligned}\neg\,(f_\mathsf{T}(x_1,\ldots,x_l) \sim 0) &\equiv f_\mathsf{T}(x_1,\ldots,x_l)\ \mathsf{neg}(\sim)\,0,\\ &\quad \text{where, for } \sim \in \{\leq, <, \geq, >\} \text{ we have}\\ &\quad \mathsf{neg}(\leq) \text{ to be } >; \qquad \mathsf{neg}(<) \text{ to be } \geq;\\ &\quad \mathsf{neg}(\geq) \text{ to be } <; \qquad \mathsf{neg}(>) \text{ to be } \leq\\ \neg(x.\psi) &\equiv x.\neg(\psi)\\ \neg\,(\phi_1 \,\mathcal{U}\, \phi_2) &\equiv \neg(\phi_2)\,\mathcal{W}\,(\neg\phi_1 \wedge \neg\phi_2)\end{aligned}$$

Definition 4 (*δ-relaxation of TLTL($\mathcal{F}_\mathsf{T}$) formulae*) Let ϕ be a TLTL($\mathcal{F}_\mathsf{T}$) formula in which negations appear only on the propositional symbols. The δ relaxation of ϕ (for $\delta \geq 0$) over a closed interval J, denoted $\mathsf{rx}_J^\delta(\phi)$, is defined as:

$$
\begin{array}{l|l}
\mathsf{rx}^{\delta}_{J}(p) = p & \mathsf{rx}^{\delta}_{J}(\text{TRUE}) = \text{TRUE} \\
\mathsf{rx}^{\delta}_{J}(\neg p) = \neg p & \mathsf{rx}^{\delta}_{J}(\text{FALSE}) = \text{FALSE} \\
\mathsf{rx}^{\delta}_{J}(\phi_1 \wedge \phi_2) = \mathsf{rx}^{\delta}_{J}(\phi_1) \wedge \mathsf{rx}^{\delta}_{J}(\phi_2) & \mathsf{rx}^{\delta}_{J}(\phi_1 \vee \phi_2) = \mathsf{rx}^{\delta}_{J}(\phi_1) \vee \mathsf{rx}^{\delta}_{J}(\phi_2) \\
\mathsf{rx}^{\delta}_{J}(x.\psi) = x.\,\mathsf{rx}^{\delta}_{J}(\psi) & \\
\mathsf{rx}^{\delta}_{J}(\phi_1\,\mathcal{U}\,\phi_2) = \mathsf{rx}^{\delta}_{J}(\phi_1)\,\mathcal{U}\,\mathsf{rx}^{\delta}_{J}(\phi_2) & \mathsf{rx}^{\delta}_{J}(\phi_1\,\mathcal{W}\,\phi_2) = \mathsf{rx}^{\delta}_{J}(\phi_1)\,\mathcal{W}\,\mathsf{rx}^{\delta}_{J}(\phi_2)
\end{array}
$$

$$
\mathsf{rx}^{\delta}_{J}\left(f_{\mathsf{T}}(x_1,\ldots,x_l)) \sim 0\right) = \begin{cases} f_{\mathsf{T}}(x_1,\ldots,x_l) + K^{f_{\mathsf{T}}}_{J}(\delta) \sim 0 \text{ if } \sim \in \{>,\geq\} \\ f_{\mathsf{T}}(x_1,\ldots,x_l) - K^{f_{\mathsf{T}}}_{J}(\delta) \sim 0 \text{ if } \sim \in \{<,\leq\}, \end{cases}
$$

where $K^{f_{\mathsf{T}}}_{J} : [0, \max \mathsf{tdom}(J) - \min \mathsf{tdom}(J)] \to \mathbb{R}_+$, and

$$
K^{f_{\mathsf{T}}}_{J}(\delta) \quad \sup_{\substack{t_1,\ldots,t_l \in J \\ t'_1,\ldots,t'_l \in J}} \left\{ \left| f_{\mathsf{T}}(t_1,\ldots,t_l) - f_{\mathsf{T}}(t'_1,\ldots,t'_l) \right| \text{ s.t. } |t_i - t'_i| \leq \delta \text{ for all } i \right\} \tag{11}
$$

Thus, instead of comparing the $f_{\mathsf{T}}()$ values to 0, we relax by comparing instead to $\pm K^{f_{\mathsf{T}}}_{J}(\delta)$. The other cases recursively relax the subformulae. The functions $K^{f_{\mathsf{T}}}_{J}(\delta)$ define the maximal change in the value of f_{T} that can occur when the input variables can vary by δ. The role of J is to restrict the domain of the freeze quantifier variables to the time interval J (from $\mathbb{R}_+$) in order to obtain the least possible relaxation on a given trace π (e.g., we do not care about the values of a function in $\mathcal{F}_{\mathsf{T}}$ outside of the domain $\mathsf{tdom}(\pi)$ of the trace).

Proposition 2 *The function* rx *is a relaxation on* TLTL($\mathcal{F}_{\mathsf{T}}$) *formulae, i.e. if a timed propositional trace* $\pi \models \phi$ *for a* TLTL($\mathcal{F}_{\mathsf{T}}$) *formula* ϕ, *then* $\pi \models \mathsf{rx}^{\delta}_{J}(\phi)$ *for all* $\delta > 0$ *and non-empty intervals* J.

Proof Observe that, over the predicates $f_{\mathsf{T}}(x_1,\ldots,x_l) \sim 0$, the function rx is indeed a relaxation, *i.e*, if $f_{\mathsf{T}}(t_1,\ldots,t_l) \sim 0$ for values $t_1,\ldots,t_l$, then $\mathsf{rx}^{\delta}_{J}(f_{\mathsf{T}}(t_1,\ldots,t_l)) \sim 0)$ also holds. The result follows by a straightforward induction argument. □

Example 5 (*δ-relaxation for bounded temporal operators—MTL*) We demonstrate how δ-relaxation operates on bounded time constraints through an example. Consider the MTL formula $\phi = Q\,\mathcal{U}_{[a,b]}R$. The δ-relaxation of this formula over the interval $\mathbb{R}_+$ is $Q\,\mathcal{U}_{[a-2\cdot\delta\,,\,b+2\cdot\delta]}R$. This can be seen as follows. The formula ϕ can be written in TLTL syntax as:

$$
x.Q\,\mathcal{U}\,y.\,((y \leq x+b) \wedge (y \geq x+a) \wedge R)\,.
$$

The δ-relaxation of this formula according to Definition 4 is:

$$
\begin{aligned}
&\mathsf{rx}^{\delta}_{\mathbb{R}_+}\,(x.Q\,\mathcal{U}\,y.\,((y \leq x+b) \wedge (y \geq x+a) \wedge R)) \\
&\qquad = \mathsf{rx}^{\delta}_{\mathbb{R}_+}\,(x.Q\,\mathcal{U}\,y.\,((y-x-b \leq 0) \wedge (y-x-a \geq 0) \wedge R)) \\
&\qquad = x.Q\,\mathcal{U}\,y.\begin{pmatrix}(y-x-b-2\cdot\delta \leq 0) \wedge \\ (y-x-a+2\cdot\delta \geq 0) \wedge R\end{pmatrix} \\
&\qquad\qquad \text{since the Lipschitz constant of } y-x-c \text{ is 2} \\
&\qquad\qquad \text{(for any constant } c\text{) for the } L_\infty \text{ norm.} \\
&\qquad = x.Q\,\mathcal{U}\,y.\,((y \leq x+b+2\cdot\delta) \wedge (y \geq x+a-2\cdot\delta) \wedge R) \\
&\qquad = Q\,\mathcal{U}_{[a-2\cdot\delta,b+2\cdot\delta]}R.
\end{aligned}
$$

Thus, the time constraint interval boundaries are relaxed by $2{\cdot}\delta$. The factor of 2 arises because there are two contributing factors: the starting time of Q can be "pulled back" by δ, and the time of R can be postponed by δ; thus, the time duration in between Q and R can increase (and similarly can decrease) by $2{\cdot}\delta$. □

Theorem 3 (Transference for propositional traces) *Let π, π' be two timed propositional traces such that $\mathcal{D}_S(\pi, \pi') < \delta$ for some finite δ. Let ϕ be a closed* TLTL($\mathcal{F}_\mathsf{T}$) *formula in negation-normal form. If $\pi \models \phi$, then $\pi' \models \mathsf{rx}^{\delta}_{I_{\pi,\pi'}}(\phi)$ where $I_{\pi,\pi'}$ is the convex hull of* $\mathsf{tdom}(\pi) \cup \mathsf{tdom}(\pi')$.

Proof Denote π' by π_2 for notational convenience. Let $\mathsf{untime}(\phi)$ be the formula where all freeze variable constraints are replaced by TRUE (*e.g.* $\mathsf{untime}(x.(Q \wedge x < 5))$ is $x.(Q \wedge \text{TRUE})$). Since $\mathcal{D}(\pi, \pi_2) < \delta$, we have that there exists a retiming $r : \mathsf{tdom}(\pi) \to \mathsf{tdom}(\pi_2)$ such that

$$\pi(t) = \pi_2(\mathsf{r}(t)). \tag{12}$$

Thus, both π and π_2 have the same untimed propositional sequence. This implies that both π and π_2 satisfy $\mathsf{untime}(\phi)$ (this can formally be shown by an induction argument). Thus, both π and π_2 satisfy the temporal operator constraints of ϕ.

We now prove the theorem statement claims concerning the freeze variable constraints. We assume WLOG that ϕ does not freeze the same variable twice. Intuitively, if $\mathcal{E}$ is an assignment to freeze variables to show $\pi \models \phi$, then to show $\pi_2 \models \mathsf{rx}^{\delta}_{I_{\pi,\pi_2}}(\phi)$, we consider the environment $\mathcal{E}_r$ is defined by $\mathcal{E}_r(x) = \mathcal{E}(r(x))$. We show that under $\mathcal{E}_r(x)$, the base constraints of ϕ involving the freeze variables are satisfied provided the base constraints are relaxed according to $\mathsf{rx}^{\delta}_{I_{\pi,\pi_2}}$. The satisfaction of non-base subformulae can then be shown by an induction argument.

Formally, the proof is as follows. The conditions of Definition 3 define a proof tree in order for $\pi \models \phi$ to hold. The nodes of the trees are 3-tuples $(\pi^t, \mathcal{E}, \psi)$ where π^t is a suffix of the trace π such that $\min \mathsf{tdom}(\pi^t) = t$, and $\mathcal{E}$ is an freeze variable environment, and ψ is a formula. The proof tree has the following properties:

1. If a node is the tuple $\left(\pi^t, \mathcal{E}, \psi\right)$, then $\pi^t \models_{\mathcal{E}} \psi$.
2. A node $\left(\pi^{t_p}, \mathcal{E}_p, \psi_p\right)$ has the following children (mirroring the proof obligations for showing $\pi^{t_p} \models_{\mathcal{E}_p} \psi_p$ according to Definition 3:
   (a) if $\psi_p = \psi_1 \wedge \psi_2$, then two children nodes labelled with $\left(\pi^{t_p}, \mathcal{E}_p, \psi_1\right)$ and $\left(\pi^{t_p}, \mathcal{E}_p, \psi_2\right)$ respectively.
   (b) if $\psi_p = \psi_1 \vee \psi_2$, then one child node labelled with either $\left(\pi^{t_p}, \mathcal{E}_p, \psi_1\right)$, or $\left(\pi^{t_p}, \mathcal{E}_p, \psi_2\right)$.
   (c) If $\psi_p = \psi_1 \,\mathcal{U}\, \psi_2$, then, for a single t with $\max \mathsf{tdom}(\pi) \geq t \geq t_p$ such that $\pi^t \models_{\mathcal{E}} \psi_2$ and $\pi^{t^*} \models_{\mathcal{E}} \psi_1 \vee \psi_2$ for all $t_p \leq t^* < t$, the following (possibly uncountably many) children:
      i. $\left(\pi^{t^*}, \mathcal{E}_p, \psi_1 \vee \psi_2\right)$
      ii. $\left(\pi^{t}, \mathcal{E}_p, \psi_2\right)$
   (d) If $\psi_p = \psi_1 \,\mathcal{W}\, \psi_2$, then children based on how how $\pi^{t_p} \models_{\mathcal{E}_p} \psi_1 \,\mathcal{W}\, \psi_2$ holds (similar to the $\mathcal{U}$ case).
   (e) If $\psi_p = x.\psi$, then one child labelled with $\left(\pi^{t_p}, \mathcal{E}_p[x := t_p], \psi\right)$.
   (f) If $\psi_p = p$ for p a proposition, or $\psi_p = f_\mathsf{T}(x_1, \ldots, x_l) \sim 0$, then the node is a leaf node such that the proposition or ψ_p holds at $\pi^{t_p}(0)$.

3. The root node is $(\pi, \mathcal{E}^*, \phi)$ where $\mathcal{E}^*$ is any environment, π is the whole trace, and ϕ is the original formula.

Note that if a non-root node in a tree is labelled $\left(\pi^t, \mathcal{E}, \psi\right)$, then t is not smaller than any value $\mathcal{E}(x)$ in that tree for x free in ψ (intuitively, the variable x was bound to a value at a time earlier than, or equal to time t).

Let $\mathcal{T}$ be a proof tree for $\pi \models \phi$. We construct another proof tree $\mathcal{T}'$ which witnesses $\pi_2 \models \mathsf{rx}^{\delta}_{I_{\pi,\pi_2}}(\phi)$ as follows.

- For every node $(\pi^t, \mathcal{E}, \psi)$ of $\mathcal{T}$, the tree $\mathcal{T}'$ has a corresponding node $\left(\pi_2^{r(t)}, \mathcal{E}_r, \mathsf{rx}^{\delta}_{I_{\pi,\pi_2}}(\psi)\right)$, where the environment $\mathcal{E}_r$ is defined by $\mathcal{E}_r(x) = \mathcal{E}(r(x))$.
- Moreover, if $\left(\pi^{t_p}, \mathcal{E}_p, \psi_p\right)$ is a parent to $\left(\pi^t, \mathcal{E}, \psi\right)$ in tree $\mathcal{T}$, then in the tree $\mathcal{T}'$ the node $\left((\pi_2^{r(t_p)}, (\mathcal{E}_p)_r, \mathsf{rx}^{\delta}_{I_{\pi,\pi_2}}(\psi_p)\right)$ is a parent to $\left(\pi_2^{r(t)}, \mathcal{E}_r, \mathsf{rx}^{\delta}_{I_{\pi,\pi_2}}(\psi)\right)$.

Since r is strictly increasing and bijective, it can be checked that $\mathcal{T}'$ can be proved to be a proof tree by induction on the height of nodes, if we can show that for leaf nodes $\left(\pi_2^{r(t)}, \mathcal{E}_r, \psi\right)$, we have $\pi_2^{r(t)} \models_{\mathcal{E}_r} \psi$. We prove this next.

If a leaf node is $\left(\pi_2^{r(t)}, \mathcal{E}_r, p\right)$ for p a proposition, then $\pi_2^{r(t)} \models p$ since (a) r maps the time t in π to time $r(t)$ in π_2; and (b) $\mathcal{D}_{\mathcal{S}}(\pi, \pi_2) < \delta$ and δ is finite; thus $\pi_2(r(t)) = \pi(t)$; and (c) $\mathcal{T}$ must contain the corresponding leaf node $\left(\pi^t, \mathcal{E}, p\right)$ and so $\pi(t) = p$. The three previous facts imply $\pi_2(r(t)) = \pi(t) = p$; and hence $\pi_2^{r(t)} \models p$.

Consider a leaf node involving freeze variables $\left(\pi_2^{r(t)}, \mathcal{E}_r, \mathsf{rx}^{\delta}_{I_{\pi,\pi'}}(\psi)\right)$, for $\psi = f_{\mathsf{T}}(x_1, \ldots, x_l) \sim 0$. We need to show that $\mathsf{rx}^{\delta}_{I_{\pi,\pi_2}}(f_{\mathsf{T}}(u'_1, \ldots, u'_l) \sim 0)$ where $\mathcal{E}_r(x_i) = u'_i$. We show this fact as follows. The node in $\mathcal{T}$ corresponding to the $\mathcal{T}'$ node $\left(\pi_2^{r(t)}, \mathcal{E}_r, \mathsf{rx}^{\delta}_{I_{\pi,\pi_2}}(\psi)\right)$ is $\left(\pi^t,\ \mathcal{E},\ (f_{\mathsf{T}}(x_1, \ldots, x_l) \sim 0)\right)$. Since $\mathcal{T}$ is a proof tree for $\pi \models \phi$, we have that $\mathcal{E}(x_i) = u_i$ such that $f_{\mathsf{T}}(u_1, \ldots, u_l) \sim 0$. We also have that $\left|u_i - u'_i\right| < \delta$ since $r(u_i) = u'_i$; and moreover $u_i \in \mathsf{tdom}(\pi)$, and $u'_i \in \mathsf{tdom}(\pi_2)$. Thus,

$$\left|f_{\mathsf{T}}(u'_1, \ldots, u'_l) - f_{\mathsf{T}}(u_1, \ldots, u_l)\right| < K^{f_{\mathsf{T}}}_{I_{\pi,\pi'}}(\delta)\,.$$

Since $f_{\mathsf{T}}(u_1, \ldots, u_l) \sim 0$, we have that $f_{\mathsf{T}}(u'_1, \ldots, u'_l) + K^{f_{\mathsf{T}}}_{I_{\pi,\pi_2}}(\delta) \sim 0$ in case $\sim \in \{>, \geq\}$; and $f_{\mathsf{T}}(u'_1, \ldots, u'_l) - K^{f_{\mathsf{T}}}_{I_{\pi,\pi_2}}(\delta) \sim 0$ in case $\sim \in \{<, \leq\}$. Or in other words, $\mathsf{rx}^{\delta}_{I_{\pi,\pi_2}}(f_{\mathsf{T}}(u'_1, \ldots, u'_l) \sim 0)$.

Thus, the leaves of the tree $\mathcal{T}'$ satisfy property (1) of proof trees Property (1) can be seen to hold for non-leaf nodes using a bottom up argument following the structure of the proof tree $\mathcal{T}$. This completes the proof. □

Theorem 3 relaxes the freeze variables over the entire signal time-range $I_{\pi,\pi'}$; it can be strengthened by relaxing over a smaller range: if $\pi \models \phi$, and $t_1, \ldots, t_k$ are time-stamp assignments to the freeze variables $x_1, \ldots, x_k$ which witness π satisfying ϕ, then x_i only needs to be relaxed over $[t_i - \delta, t_i + \delta]$ rather than the larger interval $I_{\pi,\pi'}$. These smaller relaxation intervals for the freeze variables can be incorporated in Eq. 11. We omit this optimization for ease of presentation.

Example 6 Recall Example 4, and the formula ϕ presented in it. Suppose a trace π satisfies ϕ; and let $\mathcal{D}_S(\pi, \pi') < \delta$ (using the Skorokhod metric for propositional traces). Our transference theorem ensures that (i) π' will satisfy the same untimed formula $p \rightarrow \Diamond(q \wedge \Diamond r)$; and (ii) it gives a bound on how much the timing constraints need to be relaxed in ϕ in order to ensure satisfaction by π'; it states that π' satisfies the following relaxed formula ϕ'.

$$\pi' \models x.\left(p \rightarrow \Diamond\left(y.\left(q \wedge \Diamond\left[z.\left(r \wedge \left((y-x)^2+(z-y)^2+(z-x)^2 \leq d^\dagger\right)\right)\right]\right)\right)\right)$$

where $d^\dagger = d + 12 \cdot \delta^2 + 4\sqrt{3} \cdot \delta \cdot \sqrt{d}$.

This can be seen as follows. Since π satisfies ϕ, we must have time-stamps t_x, t_y, t_z bound to x, y, z respectively so that with these assignments, the formula ϕ is satisfied. Since π' is δ close to π, there is a retiming from π to π' such that the times t_x, t_y, t_z in π are mapped to corresponding time t'_x, t'_y, t'_z in π' such that (a) $|t_x - t'_x| \leq \delta$; and (b) $|t_y - t'_y| \leq \delta$; and (c) $|t_z - t'_z| \leq \delta$.

The sum $(t'_x - t'_y)^2 + (t'_y - t'_z)^2 + (t'_z - t'_x)^2$ is

$$\begin{aligned}
&= \Big((t'_x - t_x) + (t_x - t_y) + (t_y - t'_y)\Big)^2 + \Big((t'_y - t_y) + (t_y - t_z) + (t_z - t'_z)\Big)^2 \\
&\quad + \big((t'_z - t_z) + (t_z - t_x) + (t_x - t'_x)\big)^2 \\
&= 2\Big((t'_x - t_x)^2 + (t'_y - t_y)^2 + (t'_z - t_z)^2\Big) + (t_x - t_y)^2 + (t_y - t_z)^2 + (t_z - t_x)^2 \\
&\quad + 2\Big((t'_x - t_x)(t_x - t_y) + (t_y - t'_y)(t_x - t_y) + (t'_x - t_x)(t_y - t'_y)\Big) \\
&\quad + 2\Big((t'_y - t_y)(t_y - t_z) + (t_z - t'_z)(t_y - t_z) + (t'_y - t_y)(t_z - t'_z)\Big) \\
&\quad + 2\big((t'_z - t_z)(t_z - t_x) + (t_x - t'_x)(t_z - t_x) + (t'_z - t_z)(t_x - t'_x)\big) \\
&\leq 6\delta^2 + d + 4\delta\left|t_x - t_y\right| + 2\delta^2 + 4\delta\left|t_y - t_z\right| + 2\delta^2 + 4\delta\left|t_z - t_x\right| + 2\delta^2 \\
&= d + 12\delta^2 + 4\delta\left(\left|t_x - t_y\right| + \left|t_y - t_z\right| + \left|t_z - t_x\right|\right) \\
&\leq d + 12 \cdot \delta^2 + 4\sqrt{3} \cdot \delta \cdot \sqrt{d}
\end{aligned}$$

In the last step above, we use the inequality: $|a| + |b| + |c| \leq \sqrt{3} \cdot \sqrt{a^2 + b^2 + c^2}$ This inequality is obtained by applying the Cauchy-Schwartz inequality to the tuples $(|a|, |b|, |c|)$ and $(1, 1, 1)$. Thus, by Theorem 3,

we have

$$\pi' \models x.\left(Q \rightarrow \Diamond\left(y.\left(R \wedge \Diamond\left[z.\left(S \wedge \left((y-x)^2+(z-y)^2+(z-x)^2 \leq d^\dagger\right)\right)\right]\right)\right)\right)$$

where $d^\dagger = d + 12 \cdot \delta^2 + 4\sqrt{3} \cdot \delta \cdot \sqrt{d}$. □

3.3 Transference of FLTL properties for $\mathbb{R}^n$-valued traces

In the previous two subsections, we considered propositional traces. We now generalize to $\mathbb{R}^n$-valued traces, and freeze quantification over both time and $\mathbb{R}^n$ trace values in the logic FLTL (as compared to the propositional case where the freeze quantifiers were only over the time domain).

A *timed* $\mathbb{R}^n$*-valued trace* π is a function from a closed interval I of $\mathbb{R}_+$ to $\mathbb{R}^n$. To define the semantics of FLTL formulae over timed $\mathbb{R}^n$-valued sequences, we use booleanizing predicates $\mu : \mathbb{R}^m \rightarrow \mathbb{B}$, to transform $\mathbb{R}^n$-valued sequences into timed propositional sequences. These predicates – with the help of freeze variables – denote relationships between different times and values in the trace.

Since we also have freeze variables, FLTL with predicates is strictly more expressive than STL[6] (as in the propositional case [6]).

For ease of presentation, we assume n is fixed, i.e., we consider FLTL for $\mathbb{R}^n$-valued traces

Definition 5 (*FLTL*($\mathcal{F}$) *Syntax*) Given a set of vectors V (the freeze vectors, each vector having $n+1$ variables), and a set $\mathcal{F}$ of functions $f : \mathbb{R}^{m(n+1)} \to \mathbb{R}$ for $1 \leq m \leq M$ for some $M \geq 1$, the formulae of FLTL($\mathcal{F}$) are defined by the grammar:

$$\phi := \text{TRUE} \mid f(\overline{x}_{i_1}, \ldots, \overline{x}_{i_k}) \sim 0 \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \,\mathcal{U}\, \phi_2 \mid \overline{x}.\phi \quad \text{where}$$

$\overline{x}$ and $\overline{x}_{i_j} \in V$, and $\overline{x}_{i_j} = (x^0_{i_j}, x^1_{i_j}, \ldots, x^n_{i_j})$ (similarly, $\overline{x}$ is freeze vector of $n+1$ variables); and $\sim$ is $\leq$, $<$, $\geq$, or $>$. □

Closed FLTL($\mathcal{F}$) formulae are defined analogously to closed TLTL($\mathcal{F}_{\mathsf{T}}$) formulae. The semantics of FLTL($\mathcal{F}$) are similar to that of TLTL($\mathcal{F}_{\mathsf{T}}$) (Definition 3). The only new ingredient is that now the freeze variables are $n+1$-tuple freeze vectors. For a freeze vector $\overline{x} = (x^0, x^1, \ldots, x^n)$, we denote x^i by $\overline{x}[i]$. The component $\overline{x}[0]$ refers to the time dimension. The components $\overline{x}[i]$ for $i > 1$ refer to the i-th value dimension. The freeze quantifier "$\overline{x}.$" now binds time, and the trace values in $\mathbb{R}^n$ to $\overline{x}[0]$ and to $\overline{x}[1], \ldots, \overline{x}[n]$ respectively (the i-th dimension of the trace value is mapped to $\overline{x}[i]$).

Example 7 Consider the following property of $\mathbb{R}^2$ traces: "the first trace dimension is delayed by 4 time-units, squared, and then output as the second trace dimension". This can be expressed in FLTL($\mathcal{F}$) as:

$$\Box\, \overline{x}_1.\Big(\Diamond\, \overline{x}_2.\Big((\overline{x}_2[0] = \overline{x}_1[0] + 4) \;\wedge\; (\overline{x}_2[2] = \overline{x}_1[1] \cdot \overline{x}_1[1])\Big)\Big)$$

In the above expression, we let $a = b$ stand for $(a \leq b) \wedge (b \leq a)$. The time-delay aspect can be understood as follows: we first bind each timestamp to $\overline{x}_1[0]$. Then, we require that 4 time units after the timestamp in $\overline{x}_1[0]$, we get to a time (this timestamp is bound to $\overline{x}_2[0]$) such that the 2-nd value dimension is the square of the first value dimension at timestamp $\overline{x}_1[0]$. □

FLTL($\mathcal{F}$) *compared to the logic* STL* *of* [8,9]: The logic FLTL($\mathcal{F}$) extends STL* as follows: (1) it allows capture of time-stamps in freeze variables; (2) the functions in $\mathcal{F}$ need not be linear; and (3) the temporal operators are not attached to closed non-singular intervals.

δ *relaxation of* FLTL($\mathcal{F}$) Let $\mathbb{R}^n$ have the norm L. Let $\mathbf{J}$ be a mapping from $\{0, 1, \ldots, n\}$ to subsets of of $\mathbb{R}$ (with $\mathbf{J}(0)$ being a closed interval of $\mathbb{R}_+$). The interval $\mathbf{J}(i)$ for $i > 0$ denotes the range of the i-th trace dimension. The time-range is the interval $\mathbf{J}(0)$. The relaxation function $\mathsf{rx}^{\delta}_{\mathbf{J}}$, for the $\mathbb{R}^n$ norm L, which operates on FLTL($\mathcal{F}$) formulae is defined analogous to the relaxation function rx^{δ}_{J} in Definition 4. We omit the similar cases, and only present the new case for the predicates formed from $\mathcal{F}$ (the full definition can be found in Appendix).

$$\mathsf{rx}^{\delta}_{\mathbf{J}}\left(f(\overline{x}_1, \ldots, \overline{x}_l) \sim 0\right) = \begin{cases} f(\overline{x}_1, \ldots, \overline{x}_l) + K^f_{\mathbf{J}}(\delta) \sim 0 & \text{if } \sim \in \{>, \geq\}; \\ f(\overline{x}_1, \ldots, \overline{x}_l) - K^f_{\mathbf{J}}(\delta) \sim 0 & \text{if } \sim \in \{<, \leq\} \end{cases}$$

[6] STL is MITL enriched with booleanzing predicates without freeze variables.

where $K^f_{\mathbf{J}} : \left[0,\ \max_{0\le k\le n} |\max \mathbf{J}(k) - \min \mathbf{J}(k)|\right] \to \mathbb{R}_+$ is a function s.t.

$$K^f_{\mathbf{J}}(\delta) = \sup_{\substack{\overline{u}_i[k] \in \mathbf{J}(k);\ \overline{u}'_i[k] \in \mathbf{J}(k) \\ \text{for all } 1 \le i \le l; \\ \text{for all } 0 \le k \le n}} \left\{ \left| \begin{array}{c} f(\overline{u}_1, \ldots, \overline{u}_l) \\ - \\ f(\overline{u}'_1, \ldots, \overline{u}'_l) \end{array} \right| \text{ s.t. } \begin{array}{c} \left\| \overline{u}_i - \overline{u}'_i \right\|_{L^{\max}} \le \delta \\ \text{for all } 1 \le i \le l \end{array} \right\}$$

and where $L^{\max}$ denotes the norm:

$$\left\| \langle u^0, u^1, \ldots, u^n \rangle \right\|_{L^{\max}} = \max\left(\left| u^0 \right|, \left\| \langle u^1, \ldots, u^n \rangle \right\|_L \right).$$

The function $K^f_{\mathbf{J}}(\delta)$ define the maximal change in the value of f that can occur under the following constraints:

- the input $n+1$-ary values $\overline{u}_i$ can vary by at most δ in the $L^{\max}$ norm; and
- the time-domain is restricted to $\mathbf{J}(0)$; and
- the k-th value-domain is restricted to intervals in $\mathbf{J}(k)$.

The role of $\mathbf{J}$ in the above definition is to restrict the domains of time, and value dimensions, in order to obtain the least possible relaxation bounds on the signal constraints; as was done in Definition 4 for the freeze time variables.

Proposition 3 *The function* $\mathsf{rx}^\delta_{\mathbf{J}}$ *is a relaxation on* FLTL($\mathcal{F}$) *formulae, i.e. if a timed* $\mathbb{R}^n$*-valued trace* $\pi \models \phi$ *for a* FLTL($\mathcal{F}$) *formula* ϕ*, then* $\pi \models \mathsf{rx}^\delta_{\mathbf{J}}(\phi)$.

Proof The proof is similar to the proof of Proposition 2. □

Theorem 4 (Transference for $\mathbb{R}^n$-valued Traces) *Let* π, π' *be two* $\mathbb{R}^n$*-valued traces such the Skorokhod distance between them (corresponding to the* $\mathbb{R}^n$ *norm* L*) is less than* δ *for some finite* δ*. Let* ϕ *be a closed* FLTL($\mathcal{F}$) *formula in negation-normal form. If* $\pi \models \phi$*, then* $\pi' \models \mathsf{rx}^\delta_{\mathbf{J}}(\phi)$*, where*

- $\mathbf{J}$ *be a mapping from* $\{0, 1, \ldots, n\}$ *to subsets of* $\mathbb{R}$*;*
- $\mathbf{J}(0)$ *is the convex hull of* $\mathsf{tdom}(\pi) \cup \mathsf{tdom}(\pi')$*; and*
- $\mathbf{J}(k)$ *for* $n \ge k > 0$ *is* $\{\pi(t)[k] \mid t \in \mathsf{tdom}(\pi)\} \cup \{\pi'(t)[k] \mid t \in \mathsf{tdom}(\pi')\}$*; where* $\pi(t)[k]$ *denotes the* k*-th dimensional value in* $\pi(t)$ *(and similarly for* $\pi'(t)$*); and*
- $\mathsf{rx}^\delta_{\mathbf{J}}$ *is defined with respect to the norm* $L^{\max}$.

Proof The theorem can be proved along the same lines as the proof of Theorem 3. The witnessing proof tree is constructed using the time-bindings as previously. In the propositional case, a retiming r in effect maps the proposition $\pi(t)$ to $\pi'(r(t))$; in the present case, it maps the $\mathbb{R}^n$ value $\pi(t)$ to the $\mathbb{R}^n$ value $\pi'(r(t))$ such that $\left\| \pi'(r(t)) - \pi(t) \right\|_L < \delta$. This requires us to incorporate the value distortions in $\mathbb{R}^n$ (in addition to the time distortions) when relaxing the formulae $f(\overline{x}_1, \ldots, \overline{x}_l) \sim 0$. This is handled by $\mathsf{rx}^\delta_{\mathbf{J}}$ as shown previously. The rest of the proof is as in the propositional case. □

Theorem 4 can be strengthened similar to the strengthening mentioned for Theorem 3 by relaxing the variables over smaller intervals obtained from assignments to variables which witness $\pi \models \phi$.

Example 8 (Transference) Recall Example 7. Suppose we have two traces π, π' over the time interval $[0, 100]$ with trace values in $\mathbb{R}^2$. Let the range of the first value dimension be

$[-8, 8]$ and range of the second value dimension be $[0, 64]$. Let the Skorokhod distance with respect to the $\mathbb{R}^2$ norm L_∞ be δ.

Consider the formula of Example 7. Since our traces are over a finite time-interval $[0, 100]$, we need to modify the formula of Example 7 to only talk about delay constraints till time $100 - 4 = 96$ time units as follows:

$$\square\, \overline{x}_1.\Big(\overline{x}_1[0] > 96 \vee \Diamond\, \overline{x}_2.\Big((\overline{x}_2[0] = \overline{x}_1[0] + 4) \wedge (\overline{x}_2[2] = \overline{x}_1[1] \cdot \overline{x}_1[1])\Big)\Big) \tag{13}$$

Suppose the trace π satisfies the formula above. We apply Theorem 4 to get a formula which π' satisfies as follows. Expanding the "=" constraints, we get:

$$\square\, \overline{x}_1.\left(\overline{x}_1[0] > 96 \vee \Diamond\, \overline{x}_2.\left(\begin{array}{c}(\overline{x}_2[0] \leq \overline{x}_1[0] + 4) \wedge (\overline{x}_2[0] \geq \overline{x}_1[0] + 4) \wedge \\ (\overline{x}_2[2] \leq \overline{x}_1[1] \cdot \overline{x}_1[1]) \wedge (\overline{x}_2[2] \geq \overline{x}_1[1] \cdot \overline{x}_1[1])\end{array}\right)\right) \tag{14}$$

In order to apply $\mathsf{rx}^\delta_{\mathbf{J}}$ to the above formula, we compute $\mathsf{rx}^\delta_{\mathbf{J}}$ for the following basic formulae, over $\mathbf{J}$ corresponding to the previously mentioned ranges.

- $f_1(\overline{x}_1) > 0$, for $f_1(\overline{x}) = \overline{x}_1[0] - 96$.
  We have $K^{f_1}_{\mathbf{J}}(\delta)$ to be δ (since $\left\|\overline{u}_1 - \overline{u}'_1\right\|_{L^{\max}_\infty} \leq \delta$ implies that $\left|f_1(\overline{u}_1) - f_1(\overline{u}'_1)\right| \leq \delta$. Thus, $\mathsf{rx}^\delta_{\mathbf{J}}(f_1(\overline{x}_1) > 0)$, which is defined to be $f_1(\overline{x}_1) + K^{f_1}_{\mathbf{J}}(\delta) > 0$, is equal to $\overline{x}_1[0] + \delta - 96 > 0$.
- $f_2(\overline{x}_1, \overline{x}_2) \leq 0$, for $f_2(\overline{x}_1, \overline{x}_2) = \overline{x}_2[0] - \overline{x}_1[0] - 4$.
  Since if $\left\|\overline{u}_1 - \overline{u}'_1\right\|_{L^{\max}_\infty} \leq \delta$ and if $\left\|\overline{u}_2 - \overline{u}'_2\right\|_{L^{\max}_\infty} \leq \delta$ we have

  $$\sup_{\overline{u}_1, \overline{u}_2, \overline{u}'_1, \overline{u}'_2} \left|\left(\overline{u}_2[0] - \overline{u}_1[0] - 4\right) - \left(\overline{u}'_2[0] - \overline{u}'_1[0] - 4\right)\right| \;=\; 2\delta,$$

  we get that $K^{f_2}_{\mathbf{J}}(\delta)$ to be 2δ. Thus, $\mathsf{rx}^\delta_{\mathbf{J}}(f_2(\overline{x}_1, \overline{x}_2) \leq 0)$, which is defined to be $f_2(\overline{x}_1, \overline{x}_2) - K^{f_2}_{\mathbf{J}}(\delta) \leq 0$, is equal to $\overline{x}_2[0] - \overline{x}_1[0] - 4 - 2\delta \leq 0$.
- $f_2(\overline{x}_1, \overline{x}_2) \geq 0$, for $f_2(\overline{x}_1, \overline{x}_2) = \overline{x}_2[0] - \overline{x}_1[0] - 4$.
  Using the analysis from the previous case, we get $\mathsf{rx}^\delta_{\mathbf{J}}(f_2(\overline{x}_1, \overline{x}_2) \geq 0)$ to be $\overline{x}_2[0] - \overline{x}_1[0] - 4 + 2\delta \geq 0$.
- $f_3(\overline{x}_1, \overline{x}_2) \leq 0$, for $f_3(\overline{x}_1, \overline{x}_2) = \overline{x}_2[2] - \overline{x}_1[1] \cdot \overline{x}_1[1]$.
  For vectors $\overline{u}_1, \overline{u}_2, \overline{u}'_1, \overline{u}'_2$ in $\mathbb{R}_+ \times \mathbb{R}^n$ such that

  1. $|\overline{u}_k[1]| \leq 8$ and $\left|\overline{u}'_k[1]\right| \leq 8$ for $k \in \{1, 2\}$, and
  2. $0 \leq \overline{u}_k[2] \leq 64$ and $0 \leq \overline{u}'_k[2] \leq 64$ for $k \in \{1, 2\}$, and
  3. $\left\|\overline{u}_1 - \overline{u}'_1\right\|_{L^{\max}_\infty} \leq \delta$ and
  4. $\left\|\overline{u}_2 - \overline{u}'_2\right\|_{L^{\max}_\infty} \leq \delta$,

  we have

  $$\begin{aligned}
  &\sup_{\overline{u}_1, \overline{u}_2, \overline{u}'_1, \overline{u}'_2} \left|\left(\overline{u}_2[2] - \overline{u}_1[1] \cdot \overline{u}_1[1]\right) - \left(\overline{u}'_2[2] - \overline{u}'_1[1] \cdot \overline{u}'_1[1]\right)\right| \\
  &\leq \sup_{\overline{u}_1, \overline{u}_2, \overline{u}'_1, \overline{u}'_2} \left|\overline{u}_2[2] - \overline{u}'_2[2]\right| + \sup_{\overline{u}_1, \overline{u}_2, \overline{u}'_1, \overline{u}'_2} \left|\overline{u}_1[1] \cdot \overline{u}_1[1] - \overline{u}'_1[1] \cdot \overline{u}'_1[1]\right| \\
  &\leq \delta + \sup_{\overline{u}_1, \overline{u}_2, \overline{u}'_1, \overline{u}'_2} \left|\overline{u}_1[1] \cdot \overline{u}_1[1] - \overline{u}'_1[1] \cdot \overline{u}'_1[1]\right|
  \end{aligned} \tag{15}$$

Denote $\overline{u}_1[1]$ as a, and $\overline{u}'_1[1]$ as b. To compute the supremum in the last expression, we need to obtain the following:

$$\begin{aligned} \text{maximize} \quad & \left|a^2 - b^2\right| \\ \text{subject to} \quad & -8 \leq a \leq 8 \\ & -8 \leq b \leq 8 \\ & |a - b| \leq \delta \end{aligned}$$

We have $\left|a^2 - b^2\right| = |a - b| \cdot |a + b| \leq \delta \cdot |a + b|$. Using the ranges of a, b, we get $|a + b| \leq 16$. Thus, $\left|a^2 - b^2\right| \leq 16 \cdot \delta$. Substituting back in Eq. (15), we get

$$\sup_{\overline{u}_1, \overline{u}_2, \overline{u}'_1, \overline{u}'_2} \left|\left(\overline{u}_2[2] - \overline{u}_1[1] \cdot \overline{u}_1[1]\right) - \left(\overline{u}'_2[2] - \overline{u}'_1[1] \cdot \overline{u}'_1[1]\right)\right| \leq \delta + 16 \cdot \delta = 17 \cdot \delta$$

Thus, $K_{\mathbf{J}}^{f_3}(\delta)$ is $17 \cdot \delta$. Hence, $\mathsf{rx}_{\mathbf{J}}^{\delta}(f_3(\overline{x}_1, \overline{x}_2) \leq 0)$, which is defined to be $f_3(\overline{x}_1, \overline{x}_2) - K_{\mathbf{J}}^{f_3}(\delta) \leq 0$, is equal to

$$\overline{x}_2[2] - \overline{x}_1[1] \cdot \overline{x}_1[1] - 17 \cdot \delta \leq 0.$$

– $f_3(\overline{x}_1, \overline{x}_2) \geq 0$, for $f_3(\overline{x}_1, \overline{x}_2) = \overline{x}_2[2] - \overline{x}_1[1] \cdot \overline{x}_1[1]$.
Using the analysis of the previous case, we get $\mathsf{rx}_{\mathbf{J}}^{\delta}(f_3(\overline{x}_1, \overline{x}_2) \geq 0)$ to be

$$\overline{x}_2[2] - \overline{x}_1[1] \cdot \overline{x}_1[1] + 17 \cdot \delta \geq 0.$$

Using the above facts, the relaxation $\mathsf{rx}_{\mathbf{J}}^{\delta}$ of the formula in (14) is equal to

$$\Box\, \overline{x}_1. \left(\overline{x}_1[0] > 96 - \delta \ \vee \ \Diamond\, \overline{x}_2. \left(\begin{array}{c} (\overline{x}_2[0] \leq \overline{x}_1[0] + 4 + 2\delta) \wedge \\ (\overline{x}_2[0] \geq \overline{x}_1[0] + 4 - 2\delta) \wedge \\ (\overline{x}_2[2] \leq \overline{x}_1[1] \cdot \overline{x}_1[1]) + 17\delta \wedge \\ (\overline{x}_2[2] \geq \overline{x}_1[1] \cdot \overline{x}_1[1]) - 17\delta \end{array} \right) \right)$$

This can be written in a more readable form as:

$$\Box\, \overline{x}_1. \left(\overline{x}_1[0] > 96 - \delta \ \vee \ \Diamond\, \overline{x}_2. \left(\begin{array}{c} \overline{x}_2[0] \in \left[\overline{x}_1[0]+4-2\delta,\ \overline{x}_1[0]+4+2\delta\right] \\ \wedge \\ \overline{x}_2[2] \in \left[\ (\overline{x}_1[1])^2-17\delta,\ (\overline{x}_1[1])^2+17\delta\right] \end{array} \right) \right) \tag{16}$$

Informally, the above formula specifies the following three requirements:

1. The tracking requirements are only till time $96 - \delta$ (the original formula in (13) has a tracking requirement till time 96).
2. The tracking delay is variable, and falls in the range $[4-2\delta, 4+2\delta]$ (the original formula in (13) has a tracking delay of exactly 4 time units).
3. The tracked trace dimension value v is output as the second dimension value in range $[v^2-17\delta,\ v^2+17\delta]$ (the original formula in (13) output the second dimension as exactly v^2).

Theorem 4 states that if π satisfies the formula in (13), then we can guarantee that π' satisfies the more relaxed formula in (16)

4 Quantifying timing distortion using the Skorokhod metric

In computing $\mathcal{D}_{\mathbb{S}}(\pi, \pi')$ for given traces π, π', the Skorokhod distance computation routine optimally retimes π' and then computes the pointwise value discrepancy between the retimed π' and the original π signal, i.e., it computes $\mathcal{D}_{\sup}(\pi, \pi' \circ \mathsf{r}) = \sup_{t \in I_\pi} \mathcal{D}_{\mathbb{O}}\big(\pi(t), \pi'(\mathsf{r}(t))\big)$ It is interesting to know how much retiming must be done by the routine to get the optimal Skorokhod distance. A related problem is: given a user-supplied bound ϵ, compute the least retiming r required for π' such that $\mathcal{D}_{\sup}(\pi, \pi' \circ \mathsf{r}) \leq \epsilon$. We define a measure which quantifies the retiming required as follows. Given traces π, π', and $\epsilon > 0$, let

$$\lambda^\star(\pi, \pi', \epsilon) = \begin{cases} \infty & \text{if there does not exist retiming } \mathsf{r} \\ & \text{s.t. } \mathcal{D}_{\sup}(\pi, \pi' \circ \mathsf{r}) \leq \epsilon \\ \inf\limits_{\mathsf{r} \text{ s.t. } \mathcal{D}_{\sup}(\pi, \pi' \circ \mathsf{r}) \leq \epsilon} \|\mathsf{r} - \mathcal{I}\|_{\sup} & \text{otherwise} \end{cases} \tag{17}$$

where $\|\mathsf{r} - \mathcal{I}\|_{\sup}$ is as defined in Definition 1 and quantifies the deviation of r from the identity retiming function. The quantity $\lambda^\star(\pi, \pi', \epsilon)$ above is a measure of timing distortion under an allowed value distortion ϵ. It generalizes the timing distortion quantified by the Skorokhod metric in the propositional setting. Recall that in the propositional setting (Sect. 3.1), the distance $\mathcal{D}_{\mathcal{P}}(\pi(t), \pi'(t'))$ between two trace values $\pi(t), \pi'(t')$ is 0 if $\pi(t) = \pi'(t')$, and ∞ otherwise. Under this point metric $\mathcal{D}_{\mathcal{P}}$, the Skorokhod metric between traces quantifies the timing distortion required to make the two traces match exactly. The Skorokhod distance $\mathcal{D}_{\mathbb{S}}(\pi, \pi')$ in the propositional setting is:

$$\mathcal{D}_{\mathbb{S}}(\pi, \pi') = \begin{cases} \infty & \text{if the proposition sequence of} \\ & \pi \text{ differs from that of } \pi' \\ \inf\limits_{\mathsf{r} \text{ s.t. } \mathcal{D}_{\sup}(\pi, \pi' \circ \mathsf{r}) = 0} \|\mathsf{r} - \mathcal{I}\|_{\sup} & \text{otherwise.} \end{cases}$$

The measure $\lambda^\star(\pi, \pi', \epsilon)$ of Eq. (17) generalizes the timing distortion quantification in the propositional case to the $\mathbb{R}^n$ valued case. Instead of the propositional requirement of $\mathcal{D}_{\sup}(\pi, \pi' \circ \mathsf{r}) = 0$, we require $\mathcal{D}_{\sup}(\pi, \pi' \circ \mathsf{r}) \leq \epsilon$ for a given $\epsilon > 0$.

We derive a procedure to compute $\lambda^\star(\pi, \pi', \epsilon)$ as follows. Consider a trace π_α obtained from the trace π by multiplying the timestamps by α for $\alpha > 0$. That is, $\pi_\alpha(t) = \pi(\alpha \cdot t)$. Let π'_α be defined similarly. We have $\mathcal{D}_{\mathbb{S}}\left(\pi_\alpha, \pi'_\alpha\right) < \epsilon$ iff there exists a retiming r such that

- $\|\mathsf{r} - \mathcal{I}\|_{\sup} < \epsilon$; and
- $\mathcal{D}_{\sup}(\pi_\alpha, \pi'_\alpha \circ \mathsf{r}) < \epsilon$.

Observe that r is a retiming $I_{\pi_\alpha} \to I_{\pi'_\alpha}$ between π_α and π'_α iff $\mathsf{r}_{\frac{1}{\alpha}}$ defined as $\mathsf{r}_{\frac{1}{\alpha}}(t) = \frac{\mathsf{r}(\alpha t)}{\alpha}$ is a retiming $I_\pi \to I_{\pi'}$ between π and π'. Moreover,

- $\left\|\mathsf{r}_{\frac{1}{\alpha}} - \mathcal{I}\right\|_{\sup} < \frac{\epsilon}{\alpha}$; and
- $\mathcal{D}_{\sup}(\pi, \pi' \circ \mathsf{r}_{\frac{1}{\alpha}}) < \epsilon$.

Thus, given an $\epsilon > 0$, we have

$$\lambda^\star(\pi, \pi', \epsilon) < \beta \text{ for } \beta > 0 \text{ iff } \mathcal{D}_{\mathbb{S}}\left(\pi_{\frac{\epsilon}{\beta}}, \pi'_{\frac{\epsilon}{\beta}}\right) < \epsilon.$$

The value of $\lambda^\star(\pi, \pi', \epsilon)$ can hence be found by searching for the smallest $\beta > 0$ such that $\mathcal{D}_{\mathbb{S}}\left(\pi_{\frac{\epsilon}{\beta}}, \pi'_{\frac{\epsilon}{\beta}}\right) < \epsilon$. If an upper bound T on $\lambda^\star(\pi, \pi', \epsilon)$ is given, then this can be done by

binary search over the interval $[0, T]$. Note that if $\epsilon \geq \mathcal{D}_{\sup}(\pi, \pi')$, then $\lambda^\star(\pi, \pi', \epsilon) = 0$ as no retiming will be required. If an upper bound is not given, then we set T as $(\max I_{\pi'} - \min I_\pi)$, as this is the maximum value that $\lambda^\star(\pi, \pi', \epsilon)$ can take, corresponding to the case where the starting time $\min I_\pi$ of π is mapped to the ending time $\max I_{\pi'}$ of π'. The algorithm for computing $\lambda^\star(\pi, \pi', \epsilon)$ is given below.

Algorithm 2: Computing $\lambda^\star(\pi, \pi', \epsilon)$ [defined in Eq. (17)]

Input: Traces $\pi(\cdot)$, $\pi'(\cdot)$ over times t, Bound ϵ, Maximum allowed time distortion $\tau_{\max}$, Convergence gaps α_{gap}, ϵ_{gap}
Output: Minimal retiming measure $\lambda^\star$ as defined in Eq. (17)

$d \leftarrow \mathcal{D}_{\sup}(\pi, \pi')$
if $\epsilon \geq d$ **then** $\lambda^\star \leftarrow 0$
else
  $\alpha_{lo} \leftarrow 0$
  $\alpha_{hi} \leftarrow \tau_{\max}$
  $\delta \leftarrow -\infty$
  while $\left(\alpha_{hi} - \alpha_{lo} > \alpha_{gap}\right) \wedge \left((\epsilon < \delta) \vee \left(\epsilon - \delta > \epsilon_{gap}\right)\right)$ **do**
    $\alpha \leftarrow \frac{\alpha_{hi} + \alpha_{lo}}{2}$
    $\pi, \pi' \leftarrow \pi, \pi'$ with timestamps multiplied by α
    $\delta \leftarrow \mathcal{D}_{\mathbb{S}}(\pi, \pi')$
    if $\epsilon > \delta$ **then** $\alpha_{lo} \leftarrow \alpha$
    else $\alpha_{hi} \leftarrow \alpha$
  end
end
$\lambda^\star \leftarrow \frac{\epsilon}{\alpha}$
if $\lambda^\star > \tau_{max}$ **then** $\lambda^\star \leftarrow \infty$

5 Experimental evaluation

In this section we provide experimental evidence on the efficacy of the Skorokhod metrics, and their use in a conformance testing framework. We begin the section by a brief remark on the difference between the precise Skorokhod distance between signals in a mathematical sense and the signals we obtain from a simulation framework.

Remark 1 In what follows, we rely on simulation tools that internally perform numerical integration of ordinary differential equations (ODEs) to numerically approximate the solution to the dynamical equations of a given system. A rich set of algorithms to perform such numerical integration is supported by the tools Matlab® and Simulink®. Essentially, these algorithms provide a discrete approximation of the mathematical solution to the ODEs, at a sequence of equally spaced time-points (for fixed-step solvers) or a set of variably spaced time-points (for variable-step solvers). For certain classes of variable step-solvers, the user has the ability to specify an absolute tolerance and a relative tolerance [28], which are used to bound the local truncation error or the error caused in a single iterative step in the numerical integration [33]. However, these tolerances do not control the global trun-

cation error, or the error accumulated by several integration steps. If we had access to absolute bounds on the global truncation error, to find precise bounds on the Skorokhod distance between two dynamical systems, we could find the Skorokhod distance between the discrete solution signals obtained from numerical integration, and add the global truncation error bound to the result. However, precise bounds on the global truncation error require well-behavedness conditions on the numerical integration procedure used, as well as the functions being integrated. Such conditions are nearly impossible to satisfy in practice (for example when we have hybrid systems, systems with discontinuous dynamics, black-box systems, *etc.*). Hence, in the rest of the paper, we assume the results obtained from the numerical simulation tools as the precise time-varying behavior of the systems to be analyzed.

Roadmap of the experiments In Sect. 5.1, we present the running times of our sliding window Skorokhod metric computation procedure for polygonal traces. In Sect. 5.2, we present the running times of our sliding window Skorokhod metric computation procedure when traces are completed from the sample points under a sample-and-hold semantics, and compare the distance values obtained with those given by the polygonal trace routine. This routine for piecewise constant (PWC) traces runs two orders of magnitude faster than the polygonal trace metric routine as a result of the simpler dynamic programming algorithm for PWC traces. We note that difference between the distance given by the polygonal routine, and that given by the PWC routine can be made to decrease by "up-sampling" (where we generate new intermediate sample points using linear interpolation); however, as we increase the sampling rate, the number of samples in the trace *and* the window sizes required for the same allowed time distortion *both* increase—thus up-sampling by a factor of γ results in the computation time increasing by a factor of γ^2 for PWC traces.

In Sect. 5.3, we present the results on our implementation which computes the least retiming required in order to have the $\mathcal{D}_{\text{sup}}$ distance between two given traces after retiming be at most ϵ (corresponding to the algorithm presented in Sect. 4).

In Sect. 5.4, we present three case studies for finding the distance between *systems* using the Skorokhod *trace* metric in a conformace testing framework (as sketched in Algorithm 1). The first case study involves two related LQR-based aircraft pitch controllers—the first controller is continuous time, and the second controller is in discrete time with added sensor delays. The second case study involves two air-fuel ratio controllers—the first one has highly nonlinear dynamics, and the second one is a hypothesized polynomial approximation of the first one. The third case study involves a Simulink model, under two different numerical integrators, of a four-cylinder spark ignition internal combustion engine.

5.1 Skorokhod metric computation: polygonal traces

We implemented a streaming, sliding window-based monitoring routine which checks, given a fixed δ, whether the linear interpolations of two time-sampled traces are at Skorokhod distance at most δ away from each other (the distance between two points p_1, p_2 in $\mathbb{R}^n$ is taken to be with respect to the L_2 norm, i.e., $\|p_1 - p_2\|_{L_2}$). The algortithm has a time complexity of $O(m \cdot n \cdot W)$ where m is the number of sample points, n is the dimension of a sample point, and W is the window size. Our implementation uses only $O(W)$ space. The least δ value, *i.e.* the Skorokhod distance value, is then computed by binary search over the monitoring routine. The upper limit of the search range is set to the pointwise metric (i.e., assuming the identity retiming) between the two traces.

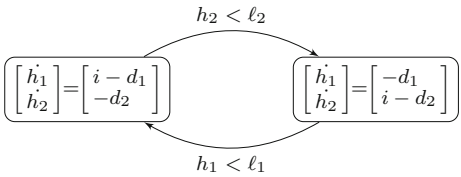

Fig. 1 System $\mathfrak{A}_1$ used for benchmarking Skorokhod Distance computation. Inflow rate i, Drain rate d_1 for tank 1 and d_2 for tank 2 are all inputs to the system

Table 1 Computation of $\mathcal{D}_{\mathbb{S}}(\pi_1, \pi_2)$, where π_1 is a trace of system $\mathfrak{A}_1$ described in Fig. 1, and π_2 is a trace of system $\mathfrak{A}_2$, which is $\mathfrak{A}_1$ with an actuation delay

Window size	Avg. $\mathcal{D}_{\mathbb{S}}$	Avg. time taken (s)		$\frac{\mathcal{D}_{\sup} - \mathcal{D}_{\mathbb{S}}}{\mathcal{D}_{\sup}}$		
		Computation	Monitoring	Max.	Avg.	SD
20	8.58	0.81	0.13	0.11	0.03	0.03
40	8.35	1.55	0.26	0.23	0.06	0.06
60	8.09	2.31	0.39	0.34	0.1	0.09
80	7.88	3.05	0.52	0.38	0.1	0.11
100	7.72	3.77	0.64	0.38	0.1	0.11

$\mathcal{D}_{\sup}$ is the pointwise trace distance with respect to the L_2 norm: $\mathcal{D}_{\sup}(\pi, \pi') = \sup_{t \in I_\pi} \|\pi(t) - \pi'(t)\|_{L_2}$. Both π_1 and π_2 contain equally spaced 2001 time points over a simulation horizon of 100 s

Time and value scaling The traces to the monitoring routine are pre-scaled, each dimension (and the time-stamp) is scaled by a different constant. The constants are chosen so that after scaling, one unit of deviation in one dimension is as undesirable as one unit of jitter in other dimensions.

Skorokhod distance computation benchmark: I We first show that the window-based implementation is efficient using the following benchmark. Figure 1 shows a hybrid dynamical system $\mathfrak{A}_1$ consisting of two water tanks, each with an outlet from which water drains at a constant rate d_j. Both tanks share a single inlet pipe that is switched between the tanks, filling only one tank at any given time at a constant inflow rate of i. When the water-level in tank j falls below level ℓ_j, the pipe switches to fill it. The drain and inflow rates d_1, d_2 and i are assumed to be inputs to the system. Now consider a version $\mathfrak{A}_2$ that incorporates an actuation delay that is a function of the inflow rate. This means that after the level drops to ℓ_j for tank j, the inlet pipe starts filling it only after a finite time. $\mathfrak{A}_1$ and $\mathfrak{A}_2$ have the same initial water level. We perform a fixed number of simulations by systematically choosing drain and inflow rates d_1, d_2, i to generate traces (water-level vs. time) of both systems and compute their Skorokhod distance. We summarize the results in Table 1.

Recall that the Skorokhod distance computation involves a sequence of monitoring calls with different δ values picked by a binary-search procedure. Thus, the total time to compute $\mathcal{D}_{\mathbb{S}}$ is the sum over the computation times for individual monitoring calls plus some bookkeeping. In Table 1, we make a distinction between the average time to monitor traces (given a δ value), and the average time to compute $\mathcal{D}_{\mathbb{S}}$. There are an average of 6 monitoring calls per $\mathcal{D}_{\mathbb{S}}$ computation. We ran 64 simulations by choosing different input values, and then computing $\mathcal{D}_{\mathbb{S}}$ for increasing window sizes. As the window size increases, the average $\mathcal{D}_{\mathbb{S}}$ decreases and the computation time increases linearly, as expected from Theorem 2. Finally, the Skorokhod distance can be significantly smaller than the simpler metric $\mathcal{D}_{\sup}$ (defined as the maximum of the pointwise L_2 norm). This discrepancy becomes more prominent with increased window size. With a window size of 100, the variation between $\mathcal{D}_{\mathbb{S}}$ and $\mathcal{D}_{\sup}$ was up to 38% (mean difference of 10% with std. deviation of 11%).

5.2 Skorokhod metric computation: piecewise constant traces

A numerical solver for simulations typically returns a sequence of time-value pairs for the signals of interest, which are then interpreted as a signal over dense time by using an interpolation scheme. The most commonly used scheme is linear interpolation, which results in polygonal, i.e., continuous piecewise linear (PWL) traces. It is also of interest to consider constant interpolation; here, the interpolated value at a time point is simply the sample-value of the largest preceding time. Recall that the Skorokhod distance computation for piecewise constant (PWC) traces can be achieved by an algorithm that uses dynamic programming, and that this algorithm is computationally more efficient than the algorithm to compute Skorokhod distance between polygonal traces. This raises a natural question: are there cases where the simpler algorithm for computing distance between PWC traces can be used? We experimentally evaluate this question by comparing the results of the polygonal routine with the results from the PWC routine.

We implemented a streaming, sliding window-based dynamic programming algorithm to compute the Skorokhod distance between two piecewise constant traces based on the results in Sect. 2.3. The algortithm has a time complexity of $O\ (m \cdot n \cdot W)$ where m is the number of sample points, n is the dimension of a sample point, and W is the window size. Our implementation uses only $O(W)$ space. The time-stamps and the sample values are pre-scaled as in the polygonal case.

Skorokhod distance computation benchmark: II In the following experiment, we asssume that we are given two models of a bang-bang controller for a water boiler system. The controller operates at a fixed frequency, and turns on heat or turns off heat depending on whether the water temperature is below or above a user-specified reference temperature. In the first model, we assume that there is a fixed actuation delay in turning the boiler on or off. In the second model, the actuation delay in turning the boiler on is different than that for turning it off. The output signal of interest is the water temperature, and we compute the distance between the outputs of the two models, for 50 randomly chosen reference temperatures in the range 40–70 °C.

We use three distance metrics to compute the distances: the $\mathcal{D}_{\text{sup}}$ metric defined in the previous benchmark, the Skorokhod distance between the output traces obtained using PWL interpolation of the samples returned by the simulator, and the Skorokhod distance between the output traces obtained using PWC interpolation. Further, we assume that the models are simulated with a variable-step solver (`ode23` in Simulink®).

For each sequence of samples returned by simulation, we compute the baseline Skorokhod distance estimation using the routine for distance estimation for PWL traces, and also compute the $\mathcal{D}_{\text{sup}}$ metric. Next, we resample the traces at increasing sampling rates and compute the Skorokhod distance using the routine for distance estimation for PWC traces. The results are shown in the table below.

Table 2 shows that the difference between the Skorokhod distance computed by the routine for PWL traces is about 15% lower than that computed by the routine for PWC traces for the same sequence of time-value pairs (i.e. sampling rate 1). This discrepancy improves as we sample the PWL traces at higher rates. However, as we increase the sampling rate, the number of samples in the trace *and* the window sizes required for the same allowed time distortion *both* increase – thus upsampling by a factor of γ results in the computation time increasing by a factor of γ^2 for PWC traces.

At around a sampling rate of 15, the computation time for the PWC routine is comparable to that of the PWL routine, while giving a discrepancy of about 1.2%. Another observation of interest is that the discrepancy between the Skorokhod distance computed by the PWC and

Table 2 Variation in Skorokhod Distance between PWC traces with sampling rate

Sampling rate	Avg. trace length	Avg. $\frac{\mathcal{D}_S^{\text{pwl}} - \mathcal{D}_S^{\text{pwc}}}{\mathcal{D}_S^{\text{pwl}}}$	Avg. $\frac{\mathcal{D}_{\text{sup}} - \mathcal{D}_S^{\text{pwc}}}{\mathcal{D}_{\text{sup}}}$	Avg. time (s)
1	381.5	−0.149	0.098	0.0034
2	761.9	−0.064	0.159	0.0104
5	1903.3	−0.033	0.181	0.0639
10	3805.6	−0.019	0.193	0.2553
15	5707.8	−0.012	0.198	0.5787
20	7610.2	−0.009	0.199	1.0365
50	19024.0	−0.003	0.204	6.5343
75	28535.5	−0.002	0.205	15.3230

The time-length of the simulation traces considered is 300 s, and we use a window corresponding to approximately 25 s of time distortion when computing the Skorokhod distances. The average trace length of the PWL (polygonal) traces was 381.5, the average time required to compute the $\mathcal{D}_S^{\text{pwl}}$ distance is 0.5225 s, while that for $\mathcal{D}_{\text{sup}}$ is 0.0001 s. The discrepancy between $\mathcal{D}_S^{\text{pwl}}$ and $\mathcal{D}_{\text{sup}}$ is 0.207

Table 3 Variation in Skorokhod Distance between polygonal traces with sampling rate

Down-sampling rate (r)	Avg. $\frac{\mathcal{D}_S^1 - \mathcal{D}_S^r}{\mathcal{D}_S^1}$	Avg. $\frac{\mathcal{D}_{\text{sup}} - \mathcal{D}_S^r}{\mathcal{D}_{\text{sup}}}$	Avg. time (s)
1	0	0.207	0.53
2	−0.159	0.092	0.28
3	−0.259	0.018	0.20
4	−0.265	0.014	0.16
5	−0.267	0.012	0.13
10	−0.274	0.007	0.08
25	−0.278	0.003	0.05
50	−0.279	0.002	0.03

We assume a window corresponding to 25 s of time distortion when computing the Skorokhod distances. $\mathcal{D}_S^r$ is the Skorokhod distance value obtain at a downsampling rate r. All data presented represents the average over 50 random simulations

the $\mathcal{D}_{\text{sup}}$ metric steadily increases with sampling rate, approaching the discrepancy between the Skorokhod distance computed by the PWL routine and the $\mathcal{D}_{\text{sup}}$ metric.

Skorokhod distance computation benchmark: III We now examine the effect on accuracy of the distance computation for the algorithm to compute Skorokhod distance on polygonal traces with sampling rate of the trace. The thesis is that as we downsample the given pair of outputs, we can compute Skorokhod distance faster, but with loss of accuracy. The experiment below (using the same setting as the previous benchmarking experiment) explores this tradeoff curve. The results are presented in Table 3.

Table 3 shows that as we down-sample the traces, the computation time improves. However, as expected, the accuracy of computation degrades. Down-sampling by a factor of 2 (i.e. dropping every other sample point) leads to an error of 15% in the distance computation.

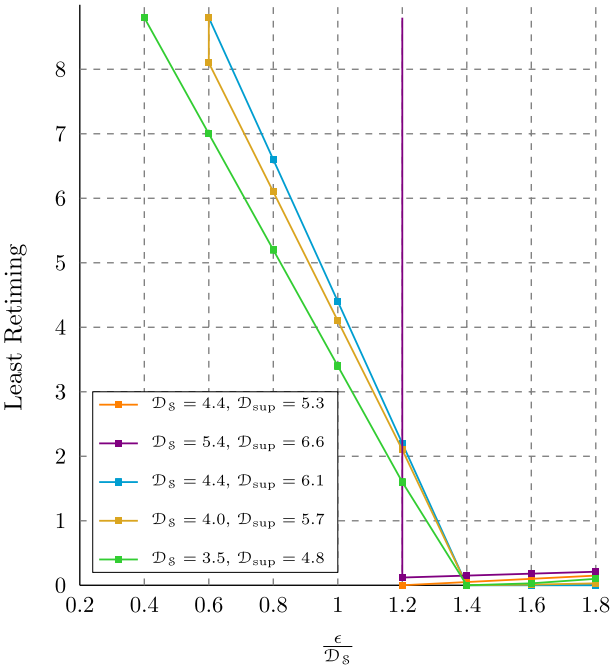

Fig. 2 Variation of least retiming required with different values of ϵ. The X-axis represents the percentage deviation of the chosen ϵ from the Skorokhod distance

5.3 Quantifying timing distortion using the Skorokhod metric

We implemented Algorithm 2 to compute the least retiming required in order to have the $\mathcal{D}_{\sup}$ distance between the traces after retiming be at most ϵ.

For each pair of traces (x_i, y_i), we compute the Skorokhod distance $\delta_i = \mathcal{D}_{\mathbb{S}}(x_i, y_i)$, and then use ϵ values in a spectrum around δ_i to explore the tradeoff between $\lambda^\star$ (as defined in Eq. (17), and the desired $\mathcal{D}_{\sup}(x \circ \mathsf{r}, y)$ bound. Figure 2 shows the results of 50 randomly simulations, where the X-axis represents the percentage deviation between the chosen ϵ and δ_i. Figure 3 is another representation of the same data, where the X-axis shows the least retiming required versus the chosen ϵ values.

For both figures, the ϵ-values were picked in the spectrum $[0.2 * \delta_i, 1.8 * \delta_i]$. As shown, there are some ϵ-values where no retiming is possible to achieve the given ϵ-bound, i.e., the retiming required is ∞. The figures also show the expected trend that as we increase the ϵ value more than the Skorokhod distance, the least retiming required to achieve the ϵ decreases, and once ϵ exceeds $\mathcal{D}_{\sup}(x, y)$, the least retiming required becomes identity (i.e., $\lambda^\star$ becomes 0).

5.4 Skorokhod distance between systems: case studies

We integrated the Skorokhod metric monitoring routine in an adaptive testing procedure for Simulink blocks based on Algorithm 1. The output of Algorithm 1 is compared against tolerance levels (e.g., maximum allowed jitter) given by the engineering requirements. In the following, we evaluate the effectiveness of the Skorokhod metric in conformance testing of

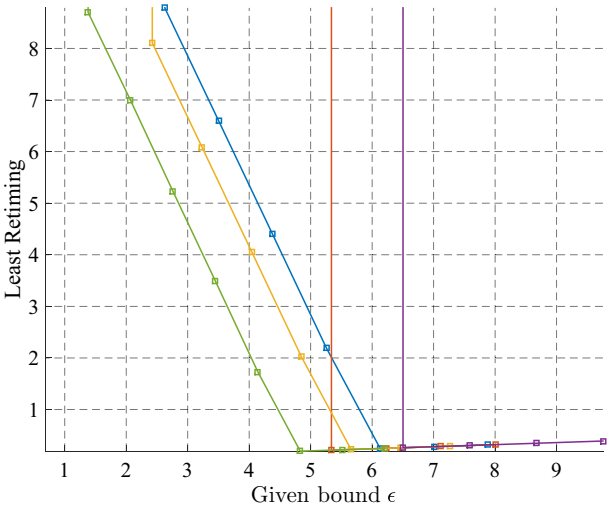

Fig. 3 Variation of least retiming measure $\lambda^\star$ required with different values of ϵ. For each trace, we chosen a different array of ϵ values for this experiment. The X-axis represents the spectrum of the chosen ϵ-values across all traces

Simulink applications. The subsequent case studies used the polygonal Skorokhod distance computation routine.

Case study I: LQR-based controller The first case study for conformance testing is an aircraft pitch control application taken from the openly accessible control tutorials for Matlab and Simulink [30]. The authors describe a linear dynamical system of the form: $\dot{\mathbf{x}} = (A - BK)\mathbf{x} + B\theta_{des}$. Here, $\mathbf{x}$ describes the vector of continuous state variables $\mathbf{x} = [\alpha\ q\ \theta]$, where α is the angle of attack, q is the pitch rate, and θ is the pitch angle. The system has a single input δ (the elevator deflection angle); and θ_{des} is the desired reference provided as an external input. In deriving the control law, the designers use the state feedback law to substitute $\delta = \theta_{des} - K\mathbf{x}$. The resulting dynamical equations of the system are of the form $\dot{\mathbf{x}} = (A - BK)\mathbf{x} + B\theta_{des}$, and the output of the system is the state variable θ. Note that the K matrix is the gain matrix resulting from the LQR control design technique. The values of the A, B and K matrices are as given below:

$$A = \begin{bmatrix} -0.313 & 56.7 & 0 \\ -0.0139 & -0.426 & 0 \\ 0 & 56.7 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0.232 \\ 0.0203 \\ 0 \end{bmatrix}$$
$$K = [-0.6435 \quad 169.6950 \quad 7.0711]$$

We are interested in studying a digital implementation of the continuous-time controller obtained using the LQR method. To do so, we consider sampled-data control where the controller samples the plant output, computes, and provides the control input to the plant every Δ seconds. To model sensor delay, we add a fixed delay element to the system; thus, the overall system now represents a delay-differential equation.

Control engineers are typically interested in the step response of a system. In particular, quantities such as the overshoot/undershoot of the output signal (maximum positive/negative deviation from a reference value) and the settling time (time it takes for transient behaviors to converge to some small region around the reference value) are of interest. Given a settling

Table 4 Variation in Skorokhod Distance with changing sampling time for an aircraft pitch control system with an LQR-based controller

Controller Sample-Time (s)	Skorokhod distance	Time taken (s) to compute $\mathcal{D}_S$	Number of simulations
0.01	0.012	232	104
0.05	0.049	96	104
0.1	0.11	70	106
0.3	0.39	45	104
0.5	1.51	40	101

Time taken indicates the total time spent in computing the upper bound on the Skorokhod distance across all simulations. We choose a window size chosen of 150 samples and simulate the system for 5 s with a variable-step solver

time and overshoot for the first system, we would like the second system to display similar characteristics. We remark that both of these properties can be expressed in STL (and hence in FLTL), see [23] for details. We quantify system conformance (and thereby adherence to requirements) in terms of the Skorokhod distance, or, in other words, maximum permitted time/space-jitter value δ. For this system, we know that at nominal conditions, the settling time is approximately 2.5 s, and that we can tolerate an increase in settling time of about 0.5 s. Thus, we chose a time-scaling factor of $2 = \frac{1}{0.5}$. We observe that the range of θ is about 0.4 radians, and specify an overshoot of 20% of this range as being permissible. Thus, we pick a scaling factor of $\frac{1}{0.08}$ for the signal domain. In other words, Skorokhod distance $\delta = 1$ corresponds to either a time-jitter of 0.5 s, or a space-discrepancy of 0.08 radians.

We summarize the results of conformance testing for different values of sampling time Δ in Table 4. As expected, the conformance increases with increasing Δ. The time taken to compute the Skorokhod distance decreases with increasing Δ, as the number of time-points in the two traces decreases.

Case study II: air-fuel ratio controller In [23], the authors present three systems representing an air-fuel ratio (λ) controller for a gasoline engine, that regulate λ to a given reference value of $\lambda_{\text{ref}} = 14.7$. These systems are simplified versions of industrial-scale models. Of interest to us are the second and the third systems. Both versions have 2 exogenous inputs, and states in both versions consist of 4 components taking values in $\mathbb{R}$ (thus, both systems have a continuous state space). The inputs are engine speed (measured in rpm) and the throttle angle (in degrees). The throttle angle is a user input, and it is common to assume a series of pulses or steps as throttle angle inputs. The engine speed is considered an input to avoid modeling parts of the powertrain dynamics. In our experiments, we typically hold the engine speed constant. This is to mimic a common engine testing scenario involving a dynamometer, which is a device to provide external torque to the engine to maintain it at a constant speed. Of the 4 state components, we assume that 2 of these are from the plant model (that encapsulates physical processes within the engine), while the other 2 belong to the controller. The plant state components p and λ denote intake manifold pressure and the A/F ratio respectively. The controller state component p_e denotes the estimated manifold pressure (with the use of an observer) used in the feed-forward control, and the state component i denotes the integrator state in the P+I feedback control. We check conformance with respect to the system output λ. For the dynamical system equations, please refer to [23,24].

The second sysrem has a continuous-time plant model with highly nonlinear dynamics, and a discrete-time controller model. In [24], the authors present a version of this system

Table 5 Conformance testing for closed-loop A/F ratio controller at different engine speeds

Engine speed (rpm)	Skorokhod distance	Computation time (s)	Total time taken (secs)	Number of simulations
1000	0.47	218	544	700
1500	0.20	240	553	700
2000	0.27	223	532	700

We scale the signals such that 0.5 s of time-jitter is treated equivalent to 10% of the steady-state value (14.7) of the A/F ratio signal. The simulation traces correspond to a time horizon of 10 s and the window size is 300

where the controller is also continuous. We take this to be $\mathfrak{A}_1$. The third system in [23] is a continuous-time closed-loop system where all the system differential equations have right-hand-sides that are polynomial approximations of the nonlinear dynamics in $\mathfrak{A}_1$. We call this polynomial dynamical system $\mathfrak{A}_2$. The rationale for these system versions is as follows: existing formal methods tools cannot reason about highly nonlinear dynamical systems, but tools such as Flow* [12], C2E2 [18], and CORA [3] demonstrate good capabilities for polynomial dynamical systems. Thus, the hope is to analyze the simpler systems instead. In [23], the authors comment that the system transformations are not accompanied by formal guarantees.

We check for conformance using the Skorokhod metric. We pick a scaling factor of 2 for the time domain, as a time-jitter of 0.5 s is the maximum deviation we wish to tolerate in the settling time, and pick $0.68 = \frac{1}{0.1*\lambda_{\text{ref}}}$ as the scaling factor for λ (which corresponds to the worst case tolerated discrepancy in the overshoot). The scaling factors transform the problem into one where a Skorokhod distance of greater than 1 is equivalent to time distortion and/or overshoot being unacceptable in $\mathfrak{A}_2$.

Table 5 summarizes the results of conformance testing for these systems. In [23], the authors shown that both the original nonlinear system and the approximate polynomial system both satisfy the STL requirement specifying a worst-case deviation 5% of the normalized air-fuel ratio $\lambda/\lambda_{\text{ref}}$ from the reference $\lambda_{\text{ref}}/\lambda_{\text{ref}}(= 1)$; that is, for μ defined as $\mu(t) = \frac{\lambda(t)-\lambda_{\text{ref}}}{\lambda_{\text{ref}}}$ we have the requirement below

$$\varphi_{\text{error}} \equiv \Box_{[\tau_s,T]}\,|\mu| < 0.05. \tag{18}$$

In our experiments we found that for the model with polynomial dynamics, the worst robustness value for φ_{error} found by falsification tools such as Breach [17] or S-TaLiro [5] is about 0.04. Thus, the polynomial dynamics model satisfies the stricter requirement $\Box_{[\tau_s,T]}\,|\mu| < 0.01$. In other words, the model satisfies the stricter requirement that the normalized air-fuel ratio error is less than 1%. The results on transference presented in this paper guarantee that as long as the largest Skorokhod distance between the two models is less than 4% of 14.7, i.e., 0.588, the model with the nonpolynomial dynamics satisfies Requirement (18).

As shown in Table 5, our conformance testing tool found the worst-case Skorokhod distance between the two models to be 0.469 at the speed of 1000 rpm.[7] Even under the worst-case assumption that the dominant contribution to the Skorokhod distance is from

[7] In the version of this paper published previously [15], this number was reported as 0.31. This discrepancy can be attributed to the random seed selected by the optimizer used to maximize the Skorokhod distance. We report the higher number in this paper, as it was the maximum value obtained after trying a number of different random seeds.

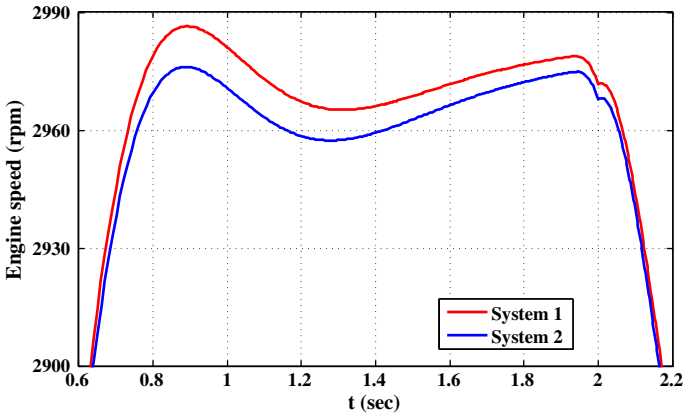

Fig. 4 Outputs showing a Skorokhod distance of 1.04

the value-domain, this means that the worst-case discrepancy between the output signals for the two models is less than $0.469\frac{1}{0.1\lambda_{\text{ref}}} = 0.31$ (which is less than the 0.588 limit postulated above). This implies that the model with nonpolynomial dynamics indeed satisfies Requirement (18).

Nevertheless, in this experiment, we found an input that led to system-outputs that differed by around 0.469 Skorokhod distance, corresponding to a worst-case value-discrepancy of roughly 2% of the reference value λ_{ref}. While this may not seem like a lot, it is about a 40% jump relative to the requirement of 5% worst-case error tolerance. This shows that though two models may satisfy the same STL requirement, it can happen that one model easily satisfies the requirement, while the other barely satisfies the requirement. Such a qualitative judgement on model conformance is valuable, and can be deduced from the quantitative conformance metric such as the one we use.

The polynomial dynamical model was obtained by approximating the nonlinear dynamics in the second model by a polynomial corresponding to minimize the error between the dynamics functions at the operating point of 1000 rpm. Hence, the largest Skorokhod distance between the models being at the input condition of 1000 rpm was surprising to the developers of these models. The designers' expectation was thus to see the two models be most conformant at 1000 rpm, and less conformant at other speeds. The root cause was this discrepancy was determined to be the increased sensitivity of the system dynamics at 1000 rpm, which led to pronounced differences in the transient behavior upon small perturbations to the dynamics. However, this was not obvious *a priori*, and was investigated only due to the unexpected results from conformance testing.

Case study III: engine timing model The Simulink demo palette from Mathworks [29] contains a system representing a four-cylinder spark ignition internal combustion engine based on a model by Crossley and Cook [13]. This system is then enhanced by adding a proportional plus integral (P + I) control law. The integrator is used to adjust the steady-state throttle as the desired engine speed set-point changes, and the proportional term compensates for phase lag introduced by the integrator. In an actual implementation of such a system, such a P+I controller is implemented using a discrete-time integrator. Such integrator blocks are typically associated with a particular numerical integration technique, e.g., forward-Euler, backward-Euler, trapezoidal, etc. It is expected that different numerical techniques will produce slight variation in the results. We wish to quantify the effect of using different numerical

integrators in a closed-loop setting. We checked if the user-provided tolerance of $\delta = 1.0$ is satisfied by systems $\mathfrak{A}_1$ and $\mathfrak{A}_2$, where $\mathfrak{A}_1$ is the original system provided in [29] and $\mathfrak{A}_2$ is a modified system that uses the backward Euler method to compute the discrete-time integral in the controller. We scale the outputs in such a way that a value discrepancy of 1% of the the output range (∼1000) is equivalent to a time discrepancy of 0.1 s. These values are chosen to bias the search towards finding signals that have a small time jitter. This is an interesting scenario for this case study where the two systems are equivalent except for the underlying numerical integration solver. We find the signal shown in Fig. 4, for which we find output traces with Skorokhod distance 1.04. The experiment uses 296 simulations and the total time taken to find the counterexample is 677 s.

6 Conclusion

We argue that the Skorokhod metric provides a robust basis for checking conformance between dynamical systems. We showed that it provides transference of a rich class of temporal logic properties and that it can be computed efficiently, both in theory and in practice. Our experiments indicate that conformance checking using the Skorokhod metric can be integrated into a testing flow for Simulink models and can find non-conformant behaviors effectively, allowing for independent weighing of time and value distortions.

Acknowledgements Open access funding provided by Max Planck Society.

Appendix

Appendix 1: Skorokod metric computation: piecewise constant traces

Proof of Lemma 1 We prove the claim as follows. Assume $I_\pi = I_{\pi'}$ contains more than one time-point (otherwise the claim is vacuous). Consider a non-decreasing retiming $\mathsf{r} : I_\pi \to I_{\pi'}$. let π and π' be constant over the intervals $[t_0, t_1), [t_1, t_2), \ldots, [t_{m-1}, t_m]$ for $t_0 = \min I_\pi$ and $t_m = \max I_\pi$.

Let I_π consist of disjoint intervals $J_0, I_0, J_1, I_1, \ldots, I_a, J_a$ (in order) such that r is constant over the I intervals, and strictly increasing over the J intervals (J_0 or J_a may be empty, but other J intervals are non-empty). Fix $\epsilon > 0$. Consider I_0. Sinces I_π and $I_{\pi'}$ contain more than one time-point, and and r is onto, at least one of J_0, J_1 is non-empty. We show that we can "wiggle" the retiming r to get r_0 such that

1. r_0 is monotone increasing over J_0, I_0, J_1 and J_k for $k \geq 2$.
2. r_0 is equal to r over $I_1, J_2, I_2, \ldots, J_a$.
3. $\|\mathsf{r}_0 - \mathsf{r}\|_{\sup} < \epsilon$ (over J_0, I_0, J_1).
4. $\pi' \circ \mathsf{r}(t) = \pi' \circ \mathsf{r}_0(t)$ for all $t \in I_\pi$, which implies that $\left\|\pi - \pi' \circ \mathsf{r}\right\|_{\sup} = \left\|\pi - \pi' \circ \mathsf{r}_0\right\|_{\sup}$.

That is, we locally perturn r a little bit so that it becomes monotone increasing over I_0, and the perturbation does not affect the trace matchings.

We obtain r_0 as follows. Denote the value of r over I_0 as t. Two cases arise.

1. $t \in (t_k, t_{k+1})$ for some k (recall the $[t_0, t_1), [t_1, t_2), \ldots, [t_{m-1}, t_m]$ breakdown of I'_π). In this case J_1 *must* be non-empty. We let r_0 be equal to r over $J_0, I_1, J_2, \ldots, J_e$. We only modify r over I_0, J_1 to get r_0.
   Consider the range of r over the interval J_1, denoted as $\mathsf{r}(J_1)$. It can be seen that r over J_1 can be modified to r_0 so that
   (a) the range of r_0 over J_1 is now $\left(t + \min(\epsilon, \frac{t_{k+1}-t}{2}),\ t_m\right] \cap \mathsf{r}(J_1)$; and
   (b) $\pi' \circ \mathsf{r}(t) = \pi' \circ \mathsf{r}_0(t)$ for all t in J_1.
   That is, we "take away" $\left(t, t + \min(\epsilon, \frac{t_{k+1}-t}{2})\right]$ from $\mathsf{r}(J_1)$. This portion can be used to make r_0 be strictly increasing on I_o, i.e., the range of r_0 on I_0 is now $\left[t, t + \min(\epsilon, \frac{t_{k+1}-t}{2})\right]$. It can be checked that r_0 satisfies the four properties listed above.
2. $t = t_k$ for some k.
   In case J_1 is non-empty, r_0 is obtained as in the previous case. In case J_1 is empty, then we have $t_k = t_m = \max I_{\pi'}$. In this condition, a portion of $\mathsf{r}(J_0)$ can be "taken away" (similar to the modification mentioned previously, to make r_0 strictly increasing over I_0 and satisfy the four conditions we desire.

Thus, for every $\epsilon > 0$, we can obtain r_0 satisfying the four conditions. Repeating the procedure, we get r_e such that

1. $\mathsf{r}_e : I_\pi \to I_{\pi'}$ is monotone strictly increasing and bijective.
2. $\|\mathsf{r}_e - \mathsf{r}\|_{\sup} < \epsilon$.
3. $\pi' \circ \mathsf{r}(t) = \pi' \circ \mathsf{r}_e(t)$ for all $t \in I_\pi$, which implies that $\left\|\pi - \pi' \circ \mathsf{r}\right\|_{\sup} = \left\|\pi - \pi' \circ \mathsf{r}_e\right\|_{\sup}$.

Thus, for every $\epsilon > 0$, given a *non-decreasing* and onto retiming r, there exists a strictly increasing and bijective retiming r_e such that

$$\max\left(\|\mathsf{r}_e - \mathsf{r}\|_{\sup},\ \left\|\pi' \circ \mathsf{r}(t) - \pi' \circ \mathsf{r}_e(t)\right\|_{\sup}\right) < \epsilon$$

This implies that the value of Eq. (1) does not change if we allow non-decreasing retimings. This complete the proof of the lemma. □

Proof of Lemma 4 Consider a non-decreasing retiming function r not in C_ϵ. To prove the claim, it suffices to show there exists $\mathsf{r}_\epsilon \in C_\epsilon$ such that

1. For all $t \in \{t_0, t_1, \ldots, t_m\}$, we have either
   - $|\mathsf{r}_\epsilon(t) - \mathsf{r}(t)| < \epsilon$, or
   - $|\mathsf{r}_\epsilon(t) - \mathcal{I}(t)| \leq |\mathsf{r}(t) - \mathcal{I}(t)|$ i.e., r_ϵ deviates less from $\mathcal{I}$ than r

   and
2. For all $t \in \{t_0, t_1, \ldots, t_m\}$, we have
   - If $r(t) = t_k$ for $t_k \in \{t_0, t_1, \ldots, t_m\}$, then $\mathsf{r}_\epsilon(t) = r(t) = t_k$; and
   - If $t_k \leq r(t) < t_{k+1}$ for $t_k \in \{t_0, t_1, \ldots, t_{m-1}\}$, then $t_k \leq \mathsf{r}_\epsilon(t) < t_{k+1}$.

   This condition can be understood as follows. The timestamps $t_0 < t_1 < \cdots < t_m$ natutally partition $[t_0, t_m]$ into intervals

$$[t_0, t_1), [t_1, t_2), \ldots, [t_{m-2}, t_{m-1})\, [t_{m-1}, t_m].$$

The stated condition says that the interval on which $\mathsf{r}_\epsilon(t)$ lies is the same as the interval on which $r(t)$ lies. Note that this implies that $\pi' \circ \mathsf{r}(t) = \pi' \circ \mathsf{r}_\epsilon(t)$ for all $t \in \{t_0, t_1, \ldots, t_m\}$.

The first condition above implies

$$\max_{t \in \{t_0, t_1, \ldots, t_m\}} |\mathsf{r}_\epsilon(t) - \mathcal{I}(t)| \;\leq\; \max_{t \in \{t_0, t_1, \ldots, t_m\}} |\mathsf{r}(t) - \mathcal{I}(t)| + \epsilon. \tag{19}$$

The second condition implies that for all $t_k \in \{t_0, t_1, \ldots, t_{m-1}\}$,

$$\begin{aligned} &\max_{t' \in \left[\mathsf{r}(t_k), \mathsf{r}(t_{k+1})\right) \cap \{t_0, t_1, \ldots, t_m\}} \mathcal{D}_{\mathcal{O}}\left(\pi(t_k),\, \pi'(t')\right) \\ &= \max_{t' \in \left[\mathsf{r}_\epsilon(t_k), \mathsf{r}_\epsilon(t_{k+1})\right) \cap \{t_0, t_1, \ldots, t_m\}} \mathcal{D}_{\mathcal{O}}\left(\pi(t_k),\, \pi'(t')\right) \end{aligned} \tag{20}$$

since $\left[\mathsf{r}(t_k), \mathsf{r}(t_{k+1})\right) \cap \{t_0, t_1, \ldots, t_m\} = \left[\mathsf{r}_\epsilon(t_k), \mathsf{r}_\epsilon(t_{k+1})\right) \cap \{t_0, t_1, \ldots, t_m\}$.

The two conditions (19) and (20) give us the statement of the lemma.

We construct r_ϵ from r as follows. We only need to specify r_ϵ on $\{t_0, t_1, \ldots, t_m\}$.

- For all t_k such that $\mathsf{r}(t_k) \in \{t_0, t_1, \ldots, t_m\}$, we let $\mathsf{r}_\epsilon(t_k) = \mathsf{r}(t_k)$.
- Suppose $\mathsf{r}(t_k) \in (t_j, t_{j+1})$. Note that we must have $k < m$ (since r is onto). Take the greatest such k.
  - If $t_k \geq t_{j+1}$ (which means t_k is closer to t_{j+1} than to t_j), then let $\mathsf{r}_\epsilon(t_k) = t_{j+1} - \epsilon$. Note that either
    - $\mathcal{I}(t_k)$ is closer to $\mathsf{r}_\epsilon(t_k)$ than to $\mathsf{r}(t_k)$ (this happens when $\mathsf{r}(t_k) < t_{j+1} - \epsilon$; or
    - $\mathsf{r}(t_k)$ and $\mathsf{r}_\epsilon(t_k)$ differ at most by ϵ) (this happens when $\mathsf{r}(t_k) \in (t_{j+1} - \epsilon, t_{j+1})$.
  - If $t_k \leq t_j$ (which means t_k is closer to t_j than to t_{j+1}), then let $\mathsf{r}_\epsilon(t_k) = t_j$. Note that $\mathcal{I}(t_k)$ is closer to $\mathsf{r}_\epsilon(t_k)$ than to $\mathsf{r}(t_k)$.

  Repeat the construction for the remaining k.

It can be checked that r_ϵ is non-decreasing, and satisfies the two conditions stated at the begining of the proof.

Appendix 2: Transference of FLTL properties for $\mathbb{R}^n$-valued traces

Definition 6 (*δ-relaxation of FLTL($\mathcal{F}$) formulae*) Let ϕ be a FLTL($\mathcal{F}$) formula in which negations appear only on the prepositional symbols . Let $\mathbf{J}$ be a mapping from $\{0, 1, \ldots, n\}$ to subsets of of $\mathbb{R}$; with $\mathbf{J}(0)$ being a closed interval of $\mathbb{R}_+$). The interval $\mathbf{J}(i)$ for $i > 0$ denotes the range of the i-th trace dimension. The time-range is the interval $\mathbf{J}(0)$. The δ relaxation of ϕ (for $\delta \geq 0$) given the interval map $jmap$, denoted $\mathsf{rx}^{\delta}_{\mathbf{J}}(\phi)$, is defined as follows (we

assume a given norm on $\mathbb{R}^n$).

$$\begin{aligned}
\mathsf{rx}^{\delta}_{\mathbf{J}}(\text{TRUE}) &= \text{TRUE}; & \mathsf{rx}^{\delta}_{\mathbf{J}}(\text{FALSE}) &= \text{FALSE};\\
\mathsf{rx}^{\delta}_{\mathbf{J}}(\phi_1 \wedge \phi_2) &= \mathsf{rx}^{\delta}_{\mathbf{J}}(\phi_1) \wedge \mathsf{rx}_{\delta}(\phi_2);\\
\mathsf{rx}^{\delta}_{\mathbf{J}}(\phi_1 \vee \phi_2) &= \mathsf{rx}^{\delta}_{\mathbf{J}}(\phi_1) \vee \mathsf{rx}^{\delta}_{\mathbf{J}}(\phi_2);\\
\mathsf{rx}^{\delta}_{\mathbf{J}}(\overline{x}.\psi) &= \overline{x}.\, \mathsf{rx}^{\delta}_{\mathbf{J}}(\psi);\\
\mathsf{rx}^{\delta}_{\mathbf{J}}(\phi_1 \,\mathcal{U}\, \phi_2) &= \mathsf{rx}^{\delta}_{\mathbf{J}}(\phi_1) \,\mathcal{U}\, \mathsf{rx}^{\delta}_{\mathbf{J}}(\phi_2);\\
\mathsf{rx}^{\delta}_{\mathbf{J}}(\phi_1 \,\mathcal{W}\, \phi_2) &= \mathsf{rx}^{\delta}_{\mathbf{J}}(\phi_1) \,\mathcal{W}\, \mathsf{rx}^{\delta}_{\mathbf{J}}(\phi_2)
\end{aligned}$$

$$\mathsf{rx}^{\delta}_{\mathbf{J}}\left(f(\overline{x}_1,\ldots,\overline{x}_l) \sim 0\right) = \begin{cases} f(\overline{x}_1,\ldots,\overline{x}_l) \;+\; K^f_{\mathbf{J}}(\delta) \;\sim\; 0 & \text{if } \sim \in \{>,\geq\};\\ f(\overline{x}_1,\ldots,\overline{x}_l) \;-\; K^f_{\mathbf{J}}(\delta) \;\sim\; 0 & \text{if } \sim \in \{<,\leq\} \end{cases}$$

where $K^f_{\mathbf{J}} : \left[0,\ \max_{0\leq k\leq n} |\max \mathbf{J}(k) \;-\; \min \mathbf{J}(k)|\right] \to \mathbb{R}_+$ is a function s.t.

$$K^f_{\mathbf{J}}(\delta) = \sup_{\substack{\overline{u}_i[k] \in \mathbf{J}(k);\ \overline{u}'_i[k] \in \mathbf{J}(k)\\ \text{for all } 1\leq i\leq l;\\ \text{for all } 0\leq k\leq n}} \left\{ \left| \begin{array}{c} f(\overline{u}_1,\ldots,\overline{u}_l)\\ - \\ f(\overline{u}'_1,\ldots,\overline{u}'_l) \end{array} \right| \text{ s.t. } \begin{array}{c} \left\|\overline{u}_i - \overline{u}'_i\right\|_{L^{\max}} \leq \delta\\ \text{for all } 1\leq i\leq l \end{array} \right\}$$

and where $L^{\max}$ denotes the norm:

$$\left\|\langle u^0, u^1, \ldots, u^n\rangle\right\|_{L^{\max}} = \max\left(\left|u^0\right|, \left\|\langle u^1,\ldots,u^n\rangle\right\|_L\right).$$

□

The function $K^f_{\mathbf{J}}(\delta)$ define the maximal change in the value of f that can occur under the following constraints:

- the input $n+1$-ary values $\overline{u}_i$ can vary by at most δ in the $L^{\max}$ norm; and
- the time-domain is restricted to $\mathbf{J}(0)$; and
- the k-th value-domain is restricted to intervals in $\mathbf{J}(k)$.

The role of $\mathbf{J}$ in the above definition is to restrict the domains of time, and value dimensions, in order to obtain the least possible relaxation bounds on the signal constraints; as was done in Definition 4 for the freeze time variables.

References

1. Abbas H, Fainekos GE (2014) Formal property verification in a conformance testing framework. In: MEMOCODE, to appear
2. Abbas H, Hoxha B, Fainekos GE, Deshmukh JV, Kapinski J, Ueda K (2014) Conformance testing as falsification for cyber-physical systems. CoRR, arXiv:1401.5200
3. Althoff M (2013) Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In: HSCC 13, pp 173–182
4. Alur R, Henzinger TA (1994) A really temporal logic. J ACM 41(1):181–204
5. Annpureddy Y, Liu C, Fainekos GE, Sankaranarayanan S (2011) S-TaLiRo: a tool for temporal logic falsification for hybrid systems. In: Proc TACAS, pp 254–257
6. Bouyer P, Chevalier F, Markey N (2005) On the expressiveness of TPTL and MTL. In: FSTTCS 05, LNCS. vol 3821, pp 432–443. Springer, Berlin

7. Branicky MS (1995) Studies in hybrid systems: modeling, analysis, and control. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA
8. Brim L, Dluhos P, Safránek D, Vejpustek T (2014) STL*: extending signal temporal logic with signal-value freezing operator. Inf Comput 236:52–67
9. Brim L, Vejpustek T, Safránek D, Fabriková J (2013) Robustness analysis for value-freezing signal temporal logic. In: Proceedings second international workshop on hybrid systems and biology, HSB 2013, EPTCS. vol 125, pp 20–36
10. Broucke M (1998) Regularity of solutions and homotopic equivalence for hybrid systems. IEEE Conf Decis Control 4:4283–4288
11. Caspi P, Benveniste A (2002) Toward an approximation theory for computerised control. In: EMSOFT, pp 294–304. Springer, Berlin
12. Chen X, Ábrahám E, Sankaranarayanan S (2013) Flow*: an analyzer for non-linear hybrid systems. CAV 13:258–263
13. Crossley PR, Cook JA (1991) A nonlinear engine model for drivetrain system development. In: International conference on control, pp 921–925. IET
14. Davoren JM (2009) Epsilon-tubes and generalized Skorokhod metrics for hybrid paths spaces. In: HSCC, LNCS. vol 5469, pp 135–149. Springer, Berlin
15. Deshmukh JV, Majumdar R, Prabhu VS (2015) Quantifying conformance using the skorokhod metric. In: Computer aided verification, CAV 2015, Part II, LNCS. vol 9207, pp 234–250. Springer, Berlin
16. Donzé A, Maler O (2010) Robust satisfaction of temporal logic over real-valued signals. In: FORMATS, LNCS. vol 6246, pp 92–106. Springer, Berlin
17. Donzé Alexandre (2010) Breach, a toolbox for verification and parameter synthesis of hybrid systems. In: CAV, pp 167–170
18. Duggirala PS, Mitra S, Viswanathan M (2013) Verification of annotated models from executions. In: EMSOFT 13, pp 26
19. Girard A, Pola G, Tabuada P (2010) Approximately bisimilar symbolic models for incrementally stable switched systems. IEEE Trans Autom Control 55(1):116–126
20. Haghverdi E, Tabuada P, Pappas GJ (2005) Bisimulation relations for dynamical, control, and hybrid systems. Theor Comput Sci 342(2–3):229–261
21. Hennessy M, Milner R (1985) Algebraic laws for nondeterminism and concurrency. J ACM 32(1):137–161
22. Henzinger MR, Henzinger TA, Kopke PW (1995) Computing simulations on finite and infinite graphs. In: FOCS: Foundations of Computer Science, pp 453–462. IEEE Computer Society
23. Jin X, Deshmukh JV, Kapinski J, Ueda K, Butts K (2014) Powertrain control verification benchmark. In: HSCC 14, pp 253–262
24. Kapinski J, Deshmukh JV, Sankaranarayanan S, Arechiga N (2014) Simulation-guided lyapunov analysis for hybrid dynamical systems. In: HSCC 14, pp 133–142. ACM, New York
25. Koymans R (1990) Specifying real-time properties with metric temporal logic. Real-Time Syst 2(4):255–299
26. Majumdar R, Prabhu VS (214) Computing the Skorokhod distance between polygonal traces (full paper). CoRR, arXiv:1410.6075
27. Majumdar R, Prabhu VS (2015) Computing the Skorokhod distance between polygonal traces. In: HSCC. ACM, New York
28. The MathWorks. https://www.mathworks.com/
29. The Mathworks. Engine timing model with closed loop control. https://www.mathworks.com/help/simulink/examples/engine-timing-model-with-closed-loop-control.html
30. Messner W, Tilbury D. Control tutorials for matlab and simulink. https://www.mathworks.com/academia/courseware/control-tutorials.html
31. Milner R (1980) A calculus of communicating systems, LNCS. vol 92, Springer, Berlin
32. Sangiorgi D, Rutten J (2011) Advanced topics in bisimulation and coinduction. cambridge University Press, Cambridge
33. Süli E, Mayers DF (2003) An introduction to numerical analysis. Cambridge University Press, Cambridge
34. Tabuada P (2009) Verification and control of hybrid systems: a symbolic approach. Springer, Berlin