Quantifying System Conformance using the Skorokhod Metric

Jyotirmoy V. Deshmukh[1], Rupak Majumdar[2], and Vinayak S. Prabhu[2]
[1] Toyota Technical Center; [2] MPI-SWS
jyotirmoy.deshmukh@tema.toyota.com {rupak,vinayak}@mpi-sws.org

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 Skorkhod 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 system conformance quantification by proving a *transference theorem*: trajectories close under the Skorokhod metric satisfy "close" logical properties. Specifically, we show the result for the timed linear time logic TLTL augmented with a rich class of temporal and spatial constraint predicates. We provide a window-based streaming algorithm to compute the Skorokhod metric, and use it 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.

1 Introduction

A fundamental question in model-based design is *conformance testing*: whether two models of a system are equivalent. For discrete systems, this question is well-studied [25, 16, 17, 26], and there is a rich theory of process equivalences based on similarity and 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," and even if two dynamical systems are bisimilar, they may still differ in many properties that are of control-theoretic interest. 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 prove results

about ϵ-bisimulation [14, 15]. 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. Thus, these notions have also been of limited practical use.

Instead, a more pragmatic semi-formal approach has gained prominence in industrial practice. In this approach, the two models are executed on the same input sequences and a metric on finite trajectories is used to evaluate how close these trajectories are. 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 an interesting class of logical 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, that the engineers consider close, as being close, and conversely.

A number of metrics have been proposed recently. The simplest is a *pointwise* metric that computes the maximum pointwise difference between two trajectories, sometimes generalized to apply a constant time-shift to one trajectory [12]. 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 report large distances for "close" models. More complicated hybrid distances have been proposed [1]. The transference properties of these metrics w.r.t. common temporal logics for dynamical systems are not yet clear.

In this work we present a methodology for quantifying conformance between real-valued dynamical systems based on the *Skorokhod* metric [11]. The Skorokhod metric allows for mismatches in both the trace values *and* in the timeline, and quantifies temporal and spatial variance 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:

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

where $\mathcal{D}$ is a pointwise metric on values. The Skorokhod distance thus 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. The Skorokhod metric was introduced as a theoretical basis for defining

the semantics of hybrid systems by providing an appropriate hybrid topology [8, 7]. 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) enriched with (i) predicates of the form $f(x_1, \ldots, x_n) \geq 0$, as in Signal Temporal Logic, for specifying constraints on trace values; and (ii) *freeze quantifiers*, as in TPTL [4], for specifying temporal constraints (freeze quantifiers can express more complex timing constraints than bounded timing constraints, *e.g* of MTL). This logic subsumes the MITL-based logic STL [12]. We prove a *transference theorem*: flows (and propositional traces) which are close under the Skorokhod metric satisfy "close" 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 [22] 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).

Usability. Using the Skorokhod distance monitoring 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 more theoretically appealing techniques such as ϵ-bisimulation function generation could not be applied. In particular, we demonstrate how two models that only differ in the underlying ODE solver can nevertheless deviate enough to invalidate system requirements on settling time.

We conclude that the Skorokhod metric can be an effective foundation for semi-formal conformance testing for complex dynamical models.

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 variance. 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 of the $(T, J, (\tau, \epsilon))$-closeness degree[1] (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

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

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 [12], in terms of robustness, deals mainly 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. The Skorokhod metric incorporates time-shifts which are variable along the timeline.

2 Preliminaries

Traces. A (finite) *trace* or a *signal* $\pi : [T_i, T_e] \mapsto \mathcal{O}$ 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 as $\mathcal{D}_{\mathcal{O}}$. The time-domain of π, denoted $\mathsf{tdom}(\pi)$ is the time domain $[T_i, T_e]$ over which it is defined. The time-duration of π, denoted as $\mathsf{tlen}(\pi)$, is $\sup(\mathsf{tdom}(\pi))$. The t-suffix of π for $t \in \mathsf{tdom}(\pi)$, denoted by π^t, is the trace π restricted to the interval $(\mathsf{tdom}(\pi) \cap [t, \mathsf{tlen}(\pi)]$. We denote by $\pi_{\downarrow T'_e}$ the prefix trace obtained from π by restricting the domain to $[T_i, T'_e] \subseteq \mathsf{tdom}(\pi)$.

Systems. A (continuous-time) *system* $\mathfrak{A} : \left(\mathbb{R}_+^{[]} \mapsto \mathcal{O}_{\mathsf{ip}}\right) \mapsto \left(\mathbb{R}_+^{[]} \mapsto \mathcal{O}_{\mathsf{op}}\right)$, where $\mathbb{R}_+^{[]}$ is the set of finite closed intervals of $\mathbb{R}_+$, transforms input traces $\pi_{\mathsf{ip}} : [T_i, T_e] \mapsto \mathcal{O}_{\mathsf{ip}}$ into output traces $\pi_{\mathsf{op}} : [T_i, T_e] \mapsto \mathcal{O}_{\mathsf{op}}$ (over the same time domain). We require that if $\mathfrak{A}(\pi_{\mathsf{ip}}) \mapsto \pi_{\mathsf{op}}$, 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 $\pi_{\mathsf{op}\downarrow T'_e}$. Thus, we only consider *causal* systems. Common examples of such systems are (causal) dynamical, and hybrid dynamical systems [6, 27].

Conformance. A system $\mathfrak{A}'$ conforms to the system $\mathfrak{A}$ over an input trace π_{ip} if $\mathfrak{A}'(\pi_{\mathsf{ip}}) = \mathfrak{A}(\pi_{\mathsf{ip}})$, *i.e.* if the behavior of $\mathfrak{A}'$ on the input trace π_{ip} is the same as that of $\mathfrak{A}$. The system $\mathfrak{A}'$ conforms to the system $\mathfrak{A}$ over the input trace set Π_{ip} if conformance holds for each input trace in Π_{ip}. Given a metric $\mathcal{D}$ over input traces, and an input trace set Π_{ip}, the *quantitative conformance* of $\mathfrak{A}'$ to $\mathfrak{A}$ over Π_{ip} is defined as the quantity $\sup_{\pi_{\mathsf{ip}} \in \Pi_{\mathsf{ip}}} \mathcal{D}\left(\mathfrak{A}'\left(\pi_{\mathsf{ip}}\right), \mathfrak{A}\left(\pi_{\mathsf{ip}}\right)\right)$. If Π_{ip} is the set of all input traces, this quantity is the analogue of the *refinement distance*.

Retimings. A *retiming* $\mathsf{r} : I \mapsto I'$, for closed intervals I, I' of $\mathbb{R}_+$ is an order preserving continuous bijective function from I to I'; thus if $t < t'$ then $\mathsf{r}(t) < \mathsf{r}(t')$. Let the class of retiming functions from I to I' be denoted as $\mathsf{R}_{I \mapsto I'}$, 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 $\mathsf{r} : I \mapsto I_\pi$; the function $\pi \circ \mathsf{r}$ is another trace from I to $\mathcal{O}$.

Definition 1 (Skorokhod Metric). *Given a retiming* $\mathsf{r} : I \mapsto I'$, *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* $\pi : I_\pi \mapsto \mathcal{O}$ *and* $\pi' : I_{\pi'} \mapsto \mathcal{O}$, *where* $\mathcal{O}$ *is a metric space with the associated metric* $\mathcal{D}_{\mathcal{O}}$, *and a*

retiming $\mathsf{r} : I_\pi \mapsto I_{\pi'}$, *let* $\|\pi \ - \ \pi' \circ \mathsf{r}\|_{\sup}$ *be defined as*

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

The Skorokhod distance[2] *between the traces* $\pi()$ *and* $\pi'()$ *is defined to be:*

$$\mathcal{D}_{\mathbb{S}}(\pi, \pi') = \inf_{r \in \mathsf{R}_{I_\pi \mapsto I_{\pi'}}} \max(\|\mathsf{r} - \mathcal{I}\|_{\sup}, \|\pi \ - \ \pi' \circ \mathsf{r}\|_{\sup}). \quad \square$$

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 value are seen can be different.

Polygonal Traces. A polygonal trace $\pi : I_\pi \mapsto \mathcal{O}$ 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 $\pi(t) = \pi(t_k) + \frac{t - t_k}{t_{k+1} - t_k} \cdot (\pi(t_{k+1}) - \pi(t_k))$. Polygonal traces are obtained when discrete-time traces are completed by linear interpolation. We remark that after retiming, the retimed trace $\pi \circ \mathsf{r}$ *need not* be piecewise linear (see [21] for an example).

Theorem 1 (Computing the Distance between Polygonal Traces [22]). *Let* $\pi : I_\pi \mapsto \mathbb{R}^n$ *and* $\pi' : I_{\pi'} \mapsto \mathbb{R}^n$ *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* $\mathcal{D}_{\mathbb{S}}(\pi, \pi')$.

1. *Given* $\delta \geq 0$, *it can be checked whether* $\mathcal{D}_{\mathbb{S}}(\pi, \pi') \leq \delta$ *in time* $O\left(m_\pi \cdot m_{\pi'} \cdot n\right)$.
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* $\mathcal{D}_{\mathbb{S}}(\pi, \pi') \leq \delta$ *in time* $O\left((m_\pi + m_{\pi'}) \cdot n \cdot W\right)$. $\square$

3 Skorokhod Distance based Conformance Testing

In conformance testing, we test for the variance in behavior of two given systems $\mathfrak{A}_1$ and $\mathfrak{A}_2$ under the same input[3]. Given the same input, the two systems produce potentially differing output traces; the goal is to quantify this difference, and to determine an input signal that causes the corresponding output signals to exceed a user provided bound on the maximum tolerable output trace distance.

Algorithm 1 is a standard optimization-guided testing algorithm in which we have used the Skorokhod distance between two output traces as the cost function. In such algorithms, it is common to define a finite parameterization of the input space, represented by the tuple (P, F, B), where $P = \{p_1, \ldots, p_k\}$

[2] 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.

[3] It is also possible to extend our approach to allow inputs that are within some bounded Skorokhod distance.

Algorithm 1: Algorithm to determine lower bound on $\max_{y_1,y_2} \mathcal{D}_{\mathbb{S}}(y_1, y_2)$

Input: System $\mathfrak{A}_1$, Model $\mathfrak{A}_2$, Bound δ, Input Parameterization (P, F, B), Time Horizon T

Output: $u(t)$ s.t. $y_1 = \mathfrak{A}_1(u)$, $y_2 = \mathfrak{A}_2(u)$, and $\mathcal{D}_{\mathbb{S}}(y_1, y_2) > \delta$

```
1  u ← random(P, F, B)
2  maxCost ← ∞, m ← 0
3  while (maxCost < δ) or (m < maxIterations) do
4  |   y1 ← simulate(M1, u, T)
5  |   y2 ← simulate(M2, u, T)
6  |   cost ← D_S(y1, y2)
7  |   if cost > maxCost then
8  |   |   cost ← maxCost
9  |   end
10 |   u ← pickNewInputs(cost)
11 |   m ← m + 1
12 end
```

represents a set of parameters, $F = \{f_1, \ldots, f_k\}$ represents a finite set of basis functions from $[0, T]$ to $\mathbb{R}^n$, where T is some finite time-horizon, and for each $p_i \in P$, there is a $b_i \in B$ that is a closed interval in $\mathbb{R}$ over which p_i is assumed to take values. An input signal u is defined such that, for all t, $u(t) = \sum_i p_i \cdot f_i(t)$. A valid input signal has the property that for all i, $p_i \in b_i$.

In each step, the algorithm picks an input signal u and computes the Skorokhod 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 current cost, and a possibly bounded history of costs, the procedure then picks a new value for u. 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.

The algorithm terminates when a violation is found (i.e., a pair of inputs that exceed the user-provided Skorokhod distance bound), or when the number of iterations is exhausted. The Skorokhod distance bound δ is chosen based on engineering requirements, *e.g.*, based on the maximum allowed weakening of the temporal logical properties that have been verified/tested on one system.

Sampling and Polygonal Approximations. In practice, the output behaviors of the systems are observed with a sampling process, thus in implementations of Algorithm 1, entities y_1 and y_2 on lines 4, 5 are time-sampled output trace *sequences*, from which the Skorokhod distance algorithm of Theorem 1 constructs (continuous time) signals using linear interpolation. Given a timed trace sequence tseq, let $[\![\mathsf{tseq}]\!]_{\mathsf{LI}}$ denote the continuous time trace obtained from tseq by linear interpolation. Let $\mathsf{tseq}_\pi, \mathsf{tseq}_{\pi'}$ be two corresponding samplings of the traces π, π'. Since the Skorokhod distance is a metric, we have that

$$\mathcal{D}_{\mathbb{S}}(\pi, \pi') \leq \mathcal{D}_{\mathbb{S}}\left([\![\mathsf{tseq}_\pi]\!]_{\mathsf{LI}}, [\![\mathsf{tseq}_{\pi'}]\!]_{\mathsf{LI}}\right) + \mathcal{D}_{\mathbb{S}}\left([\![\mathsf{tseq}_\pi]\!]_{\mathsf{LI}}, \pi\right) + \mathcal{D}_{\mathbb{S}}\left([\![\mathsf{tseq}_{\pi'}]\!]_{\mathsf{LI}}, \pi'\right).$$

If Δ_{samerr} is a bound on the distance between a trace, and an interpolated completion of its sampling, we have that $\mathcal{D}_{\mathbb{S}}(\pi, \pi') \leq \mathcal{D}_{\mathbb{S}}\left([\![\mathsf{tseq}_\pi]\!]_{\mathsf{LI}}, [\![\mathsf{tseq}_{\pi'}]\!]_{\mathsf{LI}}\right) +$

$2{\cdot}\Delta_{\mathsf{samerr}}$. Thus, in a sampling framework, a value of $2{\cdot}\Delta_{\mathsf{samerr}}$ needs to be added to the Skorokhod distance between the polygonal approximations.

Section 4 presents a theory of (quantifiable) transference of logical properties. Section 5 presents results on our implementation of Algorithm 1. We also discuss several case studies, providing rationale for choosing the appropriate δ value, and present results on the computation time and the conformance distance found.

4 Transference of Logical Properties

In this section, we demonstrate transference of logical properties. If two traces are at a distance of δ, and one trace satisfies a logical specification ϕ, we derive the "relaxation" needed (if any) in ϕ so that the other trace also satisfies this relaxed logical specification. The logic we use is a version of the timed linear time logic TLTL [4] (a timed version of the logic LTL). We show that the Skorokhod distance provides robust transference of specifications in this logic: if the Skorokhod distance between two traces is small, they satisfy close TLTL formulae. We first present the results in a propositional framework, and then extend to $\mathbb{R}^n$-valued spaces.

4.1 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: $\mathcal{D}(\sigma, \sigma') = \infty$ if $\sigma \neq \sigma'$, and 0 otherwise for $\sigma, \sigma' \in 2^{\mathcal{P}}$. We restrict our attention to propositional traces with finite variability: we require that 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. The set of all timed propositional traces over $\mathcal{P}$ is denoted by $\Pi(\mathcal{P})$.

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, the formulae of* TLTL*($\mathcal{F}_\mathsf{T}$) are defined by the following grammar.*

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

- $p \in \mathcal{P}$ *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 function, and* $\sim$ *is one of* $\{\leq, <, \geq, >\}$. □

We say that the variable x is *bound* in ϕ if ϕ is $x.\Psi$, otherwise it is *free*. The quantifier "$x.$" is known as the *freeze quantifier*, and binds the variable x to the current time. A formula is *closed* if it has no free variables.

Definition 3 (TLTL($\mathcal{F}_\mathsf{T}$) Semantics). *Let $\pi : I \mapsto 2^{\mathcal{P}}$ be a timed propositional trace, $t_0 = \min(I)$, and let $\mathcal{E} : V \mapsto I$ be the time environment mapping the variables in V to time values in I. The satisfaction of the timed sequence 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* $p \in \pi(t_0)$; $\quad \pi \models_\mathcal{E}$ 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* $z \neq x$, *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 $\pi \models_{\mathcal{E}} \phi$. $\square$

The definition of additional temporal operators in terms of these base operators is standard: 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 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.

Example 1 (Freeze quantifiers; 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 $Q\,\mathcal{U}_{\,[a,b]}R$ can be written as

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

We parse 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 R to be true, and moreover, at all times between t_x and t_y, we have $Q \vee R$ 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 $(y \leq x + b) \wedge (y \geq x + a)$. $\square$

Example 2 (Temporal Constraints). Suppose we want to express that whenever the event Q occurs, it must be followed by a response R, and then by S. In addition, we have the following timing requirement: if $\varepsilon_{QR}, \varepsilon_{RS}, \varepsilon_{QS}$ are the time delays between Q and R, between R, S, and between Q and S respectively, then: we must have $\varepsilon_{QR}^2 + \varepsilon_{RS}^2 + \varepsilon_{QS}^2 \leq d$ for a given positive constant d. This can be written using freeze quantifiers as the TLTL formula ϕ:

$$x.\,\big(Q \rightarrow \Diamond\big(y.\,\big(R \wedge \Diamond\,\big[z.\,\big(S \wedge \big((y-x)^2 + (z-y)^2 + (z-x)^2 \leq d\big)\big)\big]\big)\big)\big)\,. \quad \square$$

4.2 Transference of TLTL Properties for Propositional Traces

We show in this section 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'.

Recall that the distance between two sets of propositions σ, σ' is ∞ if $\sigma \neq \sigma'$, and 0 if $\sigma = \sigma'$. The distance between two propositional traces is defined to be the Skorokhod distance with the aforementioned metric on $2^{\mathcal{P}}$.

Next, we define relaxations of TLTL($\mathcal{F}_\mathsf{T}$)formulae. The relaxations are defined as a syntactic transformation on formulae which do not have negations, except on the prepositions. Every TLTL($\mathcal{F}_\mathsf{T}$)formula can be expressed in this negation free form. To remove negations from the until operator, we use the waiting for operator, $\mathcal{W}$, defined as:

$\pi \models_{\mathcal{E}} \phi_1\,\mathcal{W}\,\phi_2$ iff either (1) $\pi^t \models_{\mathcal{E}} \phi_1$ for all $t \in I$; or (2) $\pi^t \models_{\mathcal{E}} \phi_2$ for some $t \in I$; and $\pi^{t'} \models_{\mathcal{E}} \phi_1 \vee \phi_2$ for all $t_0 \leq t' < t$.

It can be showed that every TLTL($\mathcal{F}_\mathsf{T}$) formula can be rewritten using the $\mathcal{W}$ operator such that negations appear only over the prepositions (the procedure is given in the Appendix).

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 prepositional symbols; and let $\mathcal{F}_\mathsf{T}$ be a set of functions $f(x_1, \ldots, x_l)$ to $\mathbb{R}$, where each x_i has domain $I_{\mathcal{F}_\mathsf{T}}$ for $I_{\mathcal{F}_\mathsf{T}}$ a closed interval of $\mathbb{R}_+$. The δ relaxation of ϕ (for $\delta \geq 0$) over $I_{\mathcal{F}_\mathsf{T}}$, denoted $\mathsf{rx}^\delta_{I_{\mathcal{F}_\mathsf{T}}}(\phi)$, is defined as follows.*

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

$$
\mathsf{rx}^\delta_{I_{\mathcal{F}_\mathsf{T}}}(f_\mathsf{T}(x_1, \ldots, x_l) \sim 0) = \begin{cases} f_\mathsf{T}(x_1, \ldots, x_l) + K_{f_\mathsf{T}}(\delta) \sim 0 & \text{if } \sim \in \{>, \geq\} \\ f_\mathsf{T}(x_1, \ldots, x_l) - K_{f_\mathsf{T}}(\delta) \sim 0 & \text{if } \sim \in \{<, \leq\}, \end{cases}
$$

where $K_{f_\mathsf{T}} : [0, \max \mathsf{tdom}(I_{\mathcal{F}_\mathsf{T}}) - \min \mathsf{tdom}(I_{\mathcal{F}_\mathsf{T}})] \mapsto \mathbb{R}_+$, *and*

$$
K_{f_\mathsf{T}}(\delta) \stackrel{\text{def}}{=} \sup_{\substack{t_1, \ldots, t_l \in I_{\mathcal{F}_\mathsf{T}} \\ t'_1, \ldots, t'_l \in I_{\mathcal{F}_\mathsf{T}}}} \left\{ \left| \begin{array}{c} f_\mathsf{T}(t_1, \ldots, t_l) \\ - \\ f_\mathsf{T}(t'_1, \ldots, t'_l) \end{array} \right| \quad \text{s.t. } |t_i - t'_i| \leq \delta \text{ for all } i \right\}
$$

Thus, instead of comparing the $f_\mathsf{T}()$ values to 0, we relax by comparing instead to $\pm K_{f_\mathsf{T}}(\delta)$. The other cases are defined inductively. The functions $K_{f_\mathsf{T}}(\delta)$ define the maximal change in the value of f_T that can occur when the input variables can vary by δ. The role of $I_{\mathcal{F}_\mathsf{T}}$ is the above definition is to restrict the domain of the freeze quantifier variables to the time interval $I_{\mathcal{F}_\mathsf{T}}$ (from $\mathbb{R}_+$) in order to obtain the least possible relaxation on a given trace π (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).

Example 3. Recall Example 2, and the formula ϕ presented in it. Suppose a flow π satisfies ϕ; and let π' be δ close to π, for a finite δ, under the Skorokhod metric (for propositional traces). Our robustness theorem ensures that (i) π' will satisfy the same untimed formula $Q \rightarrow \Diamond(R \wedge \Diamond S)$; 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':

$$
x.\left(Q \rightarrow \Diamond\left(y.\left(R \wedge \Diamond\left[z.\left(S \wedge \left((y-x)^2 + (z-y)^2 + (z-x)^2 \leq d + 12{\cdot}\delta^2\right)\right)\right]\right)\right)\right). \quad \square
$$

Example 4 (δ-relaxation for Bounded Temporal Operators – MTL*).* We demonstrate how δ-relaxation operates on bounded time constraints through an example. Consider an MTL formula $\phi = Q\,\mathcal{U}_{\,[a,b]}\,R$. This can be written as a TLTL formula, and relaxed using the $\mathsf{rx}^\delta_{\mathbb{R}_+}$ function. The relaxed TLTL formula is again equivalent to an MTL formula, namely $Q\,\mathcal{U}_{\,[a-2\cdot\delta\,,\,b+2\cdot\delta]}\,R$. The details are explained in Example 8 in the Appendix. $\square$

Theorem 2 (Transference for Propositional Traces). *Let π, π' be two timed propositional traces such that $\mathcal{D}(\pi, \pi') < \delta$ for some finite δ. Let ϕ be a closed* $\mathrm{TLTL}(\mathcal{F}_\mathsf{T})$ *formula in negation free form. If $\pi \models \phi$, then $\pi' \models \mathsf{rx}^{\delta}_{I_{\pi,\pi'}}(\phi)$ where $I_{\pi,\pi'}$ is the convex hull of $\mathsf{tdom}(\pi) \cup \mathsf{tdom}(\pi')$.* □

4.3 Transference of TLTL properties for $\mathbb{R}^n$-valued Signals

A *timed $\mathbb{R}^n$-valued trace* π is a function from a closed interval I of $\mathbb{R}_+$ to $\mathbb{R}^n$. For $\overline{\alpha} = (\alpha^0, \ldots, \alpha^n) \in \mathbb{R}^n$, we denote the k-th dimensional value α^k as $\overline{\alpha}[k]$. The π projected function onto the k-th $\mathbb{R}$ dimension is denoted by $\pi_k : I \mapsto \mathbb{R}$.

In order to define the satisfaction of TLTL formulae over timed $\mathbb{R}^n$-valued sequences, we use booleanizing predicates $\mu : \mathbb{R}^n \mapsto \mathbb{B}$, as in STL [12], to transform $\mathbb{R}^n$-valued sequences in to timed propositional sequences. These predicates are part of the logical specification. In this work, we restrict out attention to traces and predicates such that each predicate varies only finitely often on the finite time traces under consideration.

Definition 5 ($\mathbf{TLTL}(\mathcal{F}_\mathsf{T}, \mathcal{F}_\mathsf{S})$ Syntax). *Given a set of variables V_T (the freeze variables), a set of* ordered *variables V_S (the signal variables), and two sets $\mathcal{F}_\mathsf{T}, \mathcal{F}_\mathsf{S}$ of functions, the formulae of* $\mathrm{TLTL}(\mathcal{F}_\mathsf{T}, \mathcal{F}_\mathsf{S})$ *are defined by the grammar:*

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

- $x \in V_\mathsf{T}$, *and* $\overline{x} = (x_1, \ldots, x_l)$ *with* $x_i \in V_\mathsf{T}$ *for all* $1 \le i \le l$;
- $\overline{y} = (y_1, \ldots, y_d)$ *with* $y_j \in V_\mathsf{S}$ *for all* $1 \le j \le d$;
- V_T *and* V_S *are disjoint;*
- $f_\mathsf{T} \in \mathcal{F}_\mathsf{T}$ *and* $f_\mathsf{S} \in \mathcal{F}_\mathsf{S}$ *are functions, and* $\sim$ *belongs to* $\{\le, <, \ge, >\}$. □

The semantics of $\mathrm{TLTL}(\mathcal{F}_\mathsf{T}, \mathcal{F}_\mathsf{S})$ is straightforward and similar to the propositional case (Definition 3). The only new ingredients are the booleanizing predicates $f_\mathsf{S}(\overline{y}) \sim 0$: we define $\pi \models_\mathcal{E} f_\mathsf{S}(y_1, \ldots, y_d) \sim 0$ iff $f_\mathsf{S}(\pi_{j_1}[t_0], \ldots, \pi_{j_d}[t_0]) \sim 0$ for any freeze variable environment $\mathcal{E}$, where $t_0 = \min \mathsf{tdom}(\pi)$, and y_i is the j_i-th variable in V_S (*i.e.*, y_i refers to the j_i-th dimension in the signal trace). We require that for a timed $\mathbb{R}^n$-valued trace π to satisfy ϕ, the arity of the functions in $\mathcal{F}_\mathsf{S}$ occurring in ϕ should not be more than n, that is, functions should not refer to dimensions greater than n for an $\mathbb{R}^n$ trace.

δ relaxation of $\mathbf{TLTL}(\mathcal{F}_\mathsf{T}, \mathcal{F}_\mathsf{S})$. Let $\mathbf{I}_{V_\mathsf{S}}$ be a mapping from V_S to closed intervals of $\mathbb{R}$ such that $\mathbf{I}_{V_\mathsf{S}}(z)$ denotes the domain of $z \in V_\mathsf{S}$. The relaxation function $\mathsf{rx}^{\delta}_{I_{\mathcal{F}_\mathsf{T}}, \mathbf{I}_{V_\mathsf{S}}}$ which operates on $\mathrm{TLTL}(\mathcal{F}_\mathsf{T}, \mathcal{F}_\mathsf{S})$ formulae is defined analogous to the relaxation function $\mathsf{rx}^{\delta}_{I_{\mathcal{F}_\mathsf{T}}}$ in Definition 4. We omit the similar cases, and only present the new case for the predicates formed from $\mathcal{F}_\mathsf{S}$ (the full definition can be found in the appendix).

$$\mathsf{rx}^{\delta}_{I_{\mathcal{F}_\mathsf{T}}, \mathbf{I}_{V_\mathsf{S}}}\left(f_\mathsf{S}(z_1, \ldots, z_l)) \sim 0\right) = \begin{cases} f_\mathsf{S}(z_1, \ldots, z_l) + K_{f_\mathsf{S}}(\delta) \sim 0 & \text{if } \sim \in \{>, \ge\}; \\ f_\mathsf{S}(z_1, \ldots, z_l) - K_{f_\mathsf{S}}(\delta) \sim 0 & \text{if } \sim \in \{<, \le\} \end{cases}$$

where $K_{f_\mathsf{S}} : \left[0,\ \max_{z \in V_\mathsf{S}} |\max \mathbf{I}_{V_\mathsf{S}}(z) - \min \mathbf{I}_{V_\mathsf{S}}(z)|\right] \mapsto \mathbb{R}_+$ is a function s.t.

$$K_{f_\mathsf{S}}(\delta) = \sup_{\substack{z_i \in \mathbf{I}_{V_\mathsf{S}}(z_i);\ z'_i \in \mathbf{I}_{V_\mathsf{S}}(z'_i) \\ \text{for all } i}} \left\{ \left| \begin{array}{c} f_\mathsf{S}(z_1, \ldots, z_l) \\ - \\ f_\mathsf{S}(z'_1, \ldots, z'_l) \end{array} \right| \text{ s.t. } |z_i - z'_i| \le \delta \text{ for all } i \right\}.$$

The functions $K_{f_{\mathsf{S}}}(\delta)$ define the maximal change in the value of f_{S} that can occur when the input variables can vary by δ. The role of $\mathbf{I}_{V_{\mathsf{S}}}$ in the above definition is to restrict the domain of the signal variables in order to obtain the least possible relaxation bounds on the signal constraints; as was done in Definition 4 for the freeze variables.

Theorem 3 (Transference for $\mathbb{R}^n$-valued Traces). *Let π, π' be two $\mathbb{R}^n$-valued traces such the Skorokhod distance between them is less than δ for some finite δ. Let ϕ be a closed* $\mathrm{TLTL}(\mathcal{F}_{\mathsf{T}}, \mathcal{F}_{\mathsf{S}})$ *formula in negation free form. If $\pi \models \phi$, then $\pi' \models \mathsf{rx}^{\delta}_{I_{\pi,\pi'}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi)$, where*

- *$I_{\pi,\pi'}$ is the convex hull of $\mathsf{tdom}(\pi) \cup \mathsf{tdom}(\pi')$; and*
- *$\mathbf{I}_{V_{\mathsf{S}}}(z)$ is the convex hull of $\{\pi(t)[k] \mid t \in \mathsf{tdom}(\pi)\} \cup \{\pi'(t)[k] \mid t \in \mathsf{tdom}(\pi')\}$; where z is the k-th variable in the ordered set V_{S}.* □

Example 5 (Spatial Constraints and Transference). Recall Example 2, suppose that the events Q, R, S are defined by the following predicates over real variables α_1 and α_2. Let $Q \equiv \alpha_1 + 10{\cdot}\alpha_2 \geq 3$; the predicate $R \equiv |\alpha_1| + |\alpha_2| \leq 20$; and $S \equiv |\alpha_1| + |\alpha_2| \leq 15$. Let π satisfy this formula with these predicates, and let π' be δ close to π, for a finite δ under the Skorokhod metric for $\mathbb{R}^2$. Our robustness theorem ensures that π' will satisfy the relaxed formula

$$x.\left(Q^{\delta} \rightarrow \Diamond\left(y.\left(R^{\delta} \wedge \Diamond\left[z.\left(S^{\delta} \wedge \left((y-x)^2 + (z-y)^2 + (z-x)^2 \leq d + 12{\cdot}\delta^2\right)\right)\right]\right)\right)\right).$$

where the relaxed predicates $Q^{\delta}, R^{\delta}, S^{\delta}$ are defined as follows: $Q^{\delta} \equiv \alpha_1 + 10{\cdot}\alpha_2 \geq 3 - 22{\cdot}\delta$; and $R^{\delta} \equiv |\alpha_1| + |\alpha_2| \leq 20 + 4{\cdot}\delta$; and $S^{\delta} \equiv |\alpha_1| + |\alpha_2| \leq 15 + 4{\cdot}\delta$. □

5 Experimental Evaluation

5.1 Skorokhod Distance Computation Benchmarking

The Skorokhod distance is computed with the help of a streaming, sliding window monitoring routine which checks for a fixed δ whether the linear interpolations of two time-sampled traces are at most δ away from each other. The least such δ value is 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. The traces to the Skorokhod procedure 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. We next present a benchmarking experiment on the distance computing routine.

Consider the hybrid dynamical system S_1 shown in Fig. 1. The system consists 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 S_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. S_1 and S_2 have the same initial

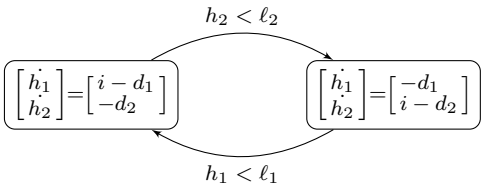

Fig. 1. System S_1 used for benchmarking Skorokhod Distance computation. Inflow rate i, Drain rate d_1 for tank 1 and d_2 for tank 2 are all inputs to the system.

Table 1. Benchmarking the computation of $\mathcal{D}_S(\pi_1, \pi_2)$, where π_1 is a trace of system S_1 described in Fig. 1, and π_2 is a trace of system S_2, which is S_1 with an actuation delay. $\mathcal{D}_2$ is the naive pointwise distance. Both π_1 and π_2 contain equally spaced 2001 time points over a simulation horizon of 100 seconds.

Window size	Avg. $\mathcal{D}_S$	Avg. Time taken (secs)		max $\frac{\mathcal{D}_2 - \mathcal{D}_S}{\mathcal{D}_2}$
		Computation	Monitoring	
20	8.58	0.81	0.13	0.09
40	8.35	1.55	0.26	0.18
60	8.09	2.31	0.39	0.26
80	7.88	3.05	0.52	0.33
100	7.72	3.77	0.64	0.38

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 $\mathcal{D}_S$ (the Skorokhod distance) computation involves a sequence of monitoring calls with different δ values picked by a bisection-search procedure. Thus, the total time to compute $\mathcal{D}_S$ 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 $\mathcal{D}_S$. There are an average of 6 monitoring calls per $\mathcal{D}_S$ computation. We ran 64 simulations by choosing different input values, and then computing $\mathcal{D}_S$ for increasing window sizes. As the window size increases, the average $\mathcal{D}_S$ is seen to decrease; this is expected as a better match may be achieved in a larger window. The computation time is also seen to increase linearly, as postulated by Theorem 1. Finally, we see that the Skorokhod distance is less aggressive at classifying traces as distant (as shown by its lower overall numbers) than a simpler metric $\mathcal{D}_2$ (defined as as the maximum of the pointwise L_2 norm[4]). We can see this discrepancy becomes more prominent with increased window size (because of better matches being available).

[4] Even though the difference is only 38% with respect to the pointwise metric, this difference is amplified in the original state value domain, as in the experiment, the input state values to the Skorokhod routine were scaled by 0.1.

5.2 Case Studies

LQR-based pitch Controller for an aircraft model The first case study is an example of an aircraft pitch control application taken from the openly accessible control tutorials for Matlab and Simulink [24]. 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 and θ_{des} is the desired reference provided as an external input. One of the states in the $\mathbf{x}$ vector represents the pitch angle θ, which is the chosen system output. The controller gain matrix K is computed using the linear quadratic regulator method [5], a standard technique from optimal control. We provide the system parameters in the appendix.

We are interested in studying the 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. Further, 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 reference value) and the settling time (time it takes for transient behaviors on the signal to converge to some small percentage of the reference value) are of interest. Given a settling 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, see [19] 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 pick δ such that the overshoot and settling time of the second system are approximately within 10% of the original system.

We summarize the results of conformance testing for different values of sampling time Δ in Table 2. It is clear that the conformance of the systems decreases with increasing Δ (which is to be expected). The time taken to compute the Skorokhod distance decreases with increasing Δ, as the number of time-points in the two traces decreases.

Air-Fuel Ratio Controller for a gasoline engine In [19], the authors present three systems representing an air-fuel ratio controller for a gasoline engine. Of interest to us are the second and the third systems. The former has a continuous-time plant model with highly nonlinear dynamics, and a discrete-time controller model. In [20], the authors present a version of this system where the controller is also continuous system. We consider this as the system S_1. The third system in [19] 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 S_1. We call this polynomial dynamical system S_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* [9], C2E2 [13] and those based on polynomial zonotopes [3] demonstrate

Table 2. Variation in Skorokhod Distance with changing sampling time for an aircraft pitch control system with an LQR-based controller. Time taken indicates the total time spent in computing the upper bound on the Skorokhod distance across all simulations. We scale the signals such that a time-jitter of 0.5 seconds, is treated the same as a value-difference of 0.08 radians, and the window size chosen is 150. The system is simulated for 5 seconds, with a variable-step solver.

Controller Sample-Time (seconds)	Skorokhod distance	Time taken (seconds) 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

Table 3. Conformance testing for closed-loop A/F ratio controller at different engine speeds. We scale the signals such that 0.5 seconds 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 seconds, and the window size is 300.

Engine speed (rpm)	Skorokhod distance	Computation Time (secs)	Total Time Taken (secs)	Number of simulations
1000	0.45	218	544	700
1500	0.20	240	553	700
2000	0.27	223	532	700

good capabilities for polynomial dynamical systems. Thus, the hope is to analyze the simpler systems instead. In [19], the authors comment that the system transformations are not accompanied by formal guarantees. By quantifying the difference in the system behaviors, we hope to show that if the system S_2 satisfies the temporal requirements φ presented in [19], then S_1 satisfies a moderate relaxation of φ. As before, we choose δ which results in an acceptable relaxation on control-theoretic requirements such as overshoot and settling time.

The results of conformance testing for these systems are summarized in Table 3. In [18], the authors posed a challenge problem for conformance testing. In this paper, the authors report that the original nonlinear system and the approximate polynomial system both satisfy the STL requirements specifying overshoot/undershoot and settling time. We, however, found an input that causes the outputs of the two systems to have a high Skorokhod distance. Thus, comparing the two systems by considering equi-satisfaction of a given set of STL requirements such as overshoot/undershoot and settling time may not always be sufficient, and our experiment indicates that the more nuanced Skorokhod metric may be a better measure of conformance.

Engine Timing Model with closed-loop control The Simulink demo palette presented by the Mathworks [23] contains a system representing a four-cylinder spark ignition internal combustion engine based on a model by Crossley and Cook [10]. This system is then enhanced by adding a proportional plus integral

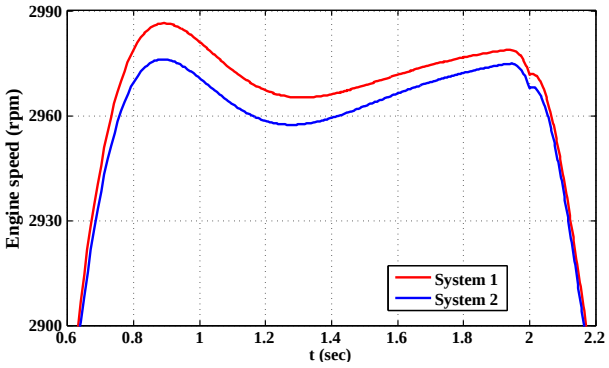

Fig. 2. Example of non-conformant behavior found using a simulation-guided optimization algorithm with the Skorokhod distance between system output trajectories as the cost function.

(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 for different numerical techniques to produce slight variation in the results, and we wish to quantify the effect of using different numerical integrators in a closed-loop setting. We try to check if the user-provided bound of $\delta = 1.0$ is satisfied by systems S_1 and S_2, where S_1 is the original system provided at [23], while S_2 is a modified system that uses the backward Euler method to compute the discrete-time integral in the controller. We try to determine the input signal that leads to a violation of this δ bound, using a simulation-guided approach as described before. We find the signal shown in Fig. 2, 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 seconds.

6 Conclusion

Metrics for comparing behaviors of dynamical systems which quantify both time and value distortions have heretofore been an object of mathematical inquiry, without enough attention being paid to computational aspects and connections to logical requirements. We argue that the Skorokhod metric provides a robust definition of conformance by proving transference of a rich class of temporal logic properties. We also demonstrate the computationally tractability of the metric for practical use by constructing a conformance testing tool in a simulation and optimization guided approach for finding and quantifying non-conformant behavior of dynamical systems. Pinpointing the source of trace deviations is necessary in many engineering applications; our tool allows for independent weighing of time and value-dimension distortions in order to achieve this objective.

References

1. H. Abbas and G.E. Fainekos. Formal property verification in a conformance testing framework. In *MEMOCODE*. To Appear, 2014.
2. H. Abbas, B. Hoxha, G.E. Fainekos, J.V. Deshmukh, J. Kapinski, and K. Ueda. Conformance testing as falsification for cyber-physical systems. *CoRR*, abs/1401.5200, 2014.
3. Matthias Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 173–182, 2013.
4. R. Alur and T.A. Henzinger. A really temporal logic. *J. ACM*, 41(1):181–204, 1994.
5. Brian D. O. Anderson. *Optimal Control: Linear Quadratic Methods*. Dover Books on Engineering. Dover Publications, 2007.
6. M.S. Branicky. *Studies in hybrid systems: modeling, analysis, and control*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.
7. M. Broucke. Regularity of solutions and homotopic equivalence for hybrid systems. In *IEEE Conference on Decision and Control*, volume 4, pages 4283–4288, Dec 1998.
8. P. Caspi and A. Benveniste. Toward an approximation theory for computerised control. In *EMSOFT*, pages 294–304. Springer, 2002.
9. Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *Computer Aided Verification*, pages 258–263, 2013.
10. PR Crossley and JA Cook. A nonlinear engine model for drivetrain system development. In *Control 1991. Control'91., International Conference on*, pages 921–925. IET, 1991.
11. J.M. Davoren. Epsilon-tubes and generalized skorokhod metrics for hybrid paths spaces. In *HSCC*, LNCS 5469, pages 135–149. Springer, 2009.
12. A. Donzé and O. Maler. Robust satisfaction of temporal logic over real-valued signals. In *FORMATS*, LNCS 6246, pages 92–106. Springer, 2010.
13. Parasara Sridhar Duggirala, Sayan Mitra, and Mahesh Viswanathan. Verification of annotated models from executions. In *Proceedings of the Eleventh ACM International Conference on Embedded Software*, page 26, 2013.
14. A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Trans. Automat. Contr.*, 55(1):116–126, 2010.
15. E. Haghverdi, P. Tabuada, and G.J. Pappas. Bisimulation relations for dynamical, control, and hybrid systems. *Theor. Comput. Sci.*, 342(2-3):229–261, 2005.
16. M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *J. ACM*, 32(1):137–161, 1985.
17. M.R. Henzinger, T.A. Henzinger, and P.W. Kopke. Computing simulations on finite and infinite graphs. In *FOCS: Foundations of Computer Science*, pages 453–462. IEEE Computer Society, 1995.
18. Xiaoqing Jin, Jyotirmoy Deshmukh, James Kapinski, Koichi Ueda, and Ken Butts. Benchmarks for model transformations and conformance checking. In *Proceedings of the 1st international workshop on Applied Verification for Continuous and Hybrid Systems (ARCH)*, 2014.
19. Xiaoqing Jin, Jyotirmoy V Deshmukh, James Kapinski, Koichi Ueda, and Ken Butts. Powertrain control verification benchmark. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pages 253–262, 2014.

20. James Kapinski, Jyotirmoy V Deshmukh, Sriram Sankaranarayanan, and Nikos Arechiga. Simulation-guided lyapunov analysis for hybrid dynamical systems. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pages 133–142. ACM, 2014.
21. R. Majumdar and V.S. Prabhu. Computing the skorokhod distance between polygonal traces (full paper). *CoRR*, abs/1410.6075, 2014.
22. R. Majumdar and V.S. Prabhu. Computing the skorokhod distance between polygonal traces. In *HSCC*. ACM, 2015.
23. The Mathworks. Engine timing model with closed loop control.
24. William Messner and Dawn Tilbury. Control tutorials for matlab and simulink.
25. R. Milner. *A Calculus of Communicating Systems*. LNCS 92. Springer, 1980.
26. D. Sangiorgi and J. Rutten. *Advanced Topics in Bisimulation and Coinduction*. Cambridge University Press, 2011.
27. P. Tabuada. *Verification and Control of Hybrid Systems - A Symbolic Approach*. Springer, 2009.

Appendix

A. Transference Formalism and Proofs

Example 6 (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

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

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

Example 7 (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 $x.\,\big(Q \to \Diamond\big(y.\,(R \wedge 0 \leq y \leq \lambda^x)\,\big)\big)$. □

Example 8 (δ-relaxation for Bounded Temporal Operators – MTL). We demonstrate how δ-relaxation operates on bounded time constraints through an example. Consider an MTL formula $\phi = Q\,\mathcal{U}_{\,[a,b]}\,R$. The δ-relaxation of this formula over the closed interval $I_{\mathcal{F}_{\mathsf{T}}} = \mathbb{R}_+$ 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:

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

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

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

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 increases by $2\cdot\delta$. □

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.

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

Proposition 1. *The function* rx *is a relaxation on* $\mathrm{TLTL}(\mathcal{F}_{\mathsf{T}})$ *formulae,* i.e. *if a timed propositional trace* $\pi \models \phi$ *for a* $\mathrm{TLTL}(\mathcal{F}_{\mathsf{T}})$ *formula* ϕ*, then* $\pi \models \mathsf{rx}^{\delta}_{I_{\mathcal{F}_{\mathsf{T}}}}(\phi)$.

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 $\mathsf{rx}^{\delta}_{I_{\mathcal{F}_{\mathsf{T}}}}\left(f_{\mathsf{T}}(t_1,\ldots,t_l)\right) \sim 0$) also holds. The result follows by a straightforward induction argument. □

Proof of Theorem 2. 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 $x.(Q \wedge \mathrm{TRUE})$). Since $\mathcal{D}(\pi,\pi') < \delta$, we have that there exists a retiming $r : \mathsf{tdom}(\pi) \mapsto \mathsf{tdom}(\pi')$ such that

$$\pi(t) = \pi'(\mathsf{r}(t)). \tag{1}$$

This implies that both π and π' satisfy $\mathsf{untime}(\phi)$, which can be shown by an induction argument. The interesting cases are for the $\mathcal{U}$ and $\mathcal{W}$ operators. We sketch the argument for the $\mathcal{U}$ case (the argument for $\mathcal{W}$ is similar). The time environment $\mathcal{E}'$ for π' assigns the time $\mathsf{r}(t_x)$ to the freeze variable x where the witnessing freeze variable environment $\mathcal{E}$ for $\pi \models \phi$ assigns t_x to x. Let $\pi \models_{\mathcal{E}} \phi_1 \,\mathcal{U}\, \phi_2$, and let t be the time value which demonstrates this satisfaction (as in Definition 3), with the corresponding freeze variable environment $\mathcal{E}$. To show $\pi' \models_{\mathcal{E}'} \phi_1 \,\mathcal{U}\, \phi_2$, we pick the time $\mathsf{r}(t)$, with the environment $\mathcal{E}'$ for π' which assigns the time $\mathsf{r}(t_x)$ to the freeze variable x where $\mathcal{E}(x) = t_x$. It can be checked that, due to Equation 1, we have (i) $\mathsf{r}(t) \geq \mathcal{E}'(x)$, for a freeze variable x in $\phi_1 \,\mathcal{U}\, \phi_2$ (which was previously bound); (i) ${\pi'}^{\mathsf{r}(t)} \models_{\mathcal{E}'} \phi_2$; and (ii) for all $t'_0 \leq t^{\dagger} < \mathsf{r}(t)$, we have ${\pi'}^{t^{\dagger}} \models_{\mathcal{E}'} \phi_1 \vee \phi_2$. Thus, $\mathsf{r}(t)$, and $\mathcal{E}'$ demonstrate that $\pi' \models_{\mathcal{E}'} \phi_1 \,\mathcal{U}\, \phi_2$.

We now check what is the relaxation needed on the original freeze variable constraints so that π' satisfies the relaxed constraints. Without loss of generality, assume that each freeze variable x is only quantified once in ϕ, *i.e.* once it is bound to a value by "$x.$", it is not "re-bound" with another application of "$x.$".

Let κ_π denote an assignment of time values (from I) to the freeze variables such that all the freeze variable constraints in ϕ are satisfied, *i.e.* κ_π is an time environment witness to the satisfaction of ϕ by π. Consider a free variable assignment $\kappa_{\pi'}$ corresponding to κ_π, where $\kappa_{\pi'}(x) = \mathsf{r}\left(\kappa_\pi(x)\right)$. This is a legal variable assignment compatible with some $\mathcal{U}$, $\mathcal{W}$ time witnesses which demonstrate

that π' satisfies $\mathsf{untime}(\phi)$, as shown previously. Observe that by the existence of a retiming function, for all freeze variables x occurring in ϕ, we have that $|\kappa_{\pi'}(x) - \kappa_\pi(x)| < \delta$.

Since the time *values* of variables are different in κ_π and $\kappa_{\pi'}$, the original freeze constraints (*e.g.* $x < 5$) in ϕ might not be satisfied with the assignment $\kappa_{\pi'}$. Consider a freeze variable constraint $f_{\mathsf{T}}(x_1, \ldots, x_l) \sim 0$ in ϕ. We know that $f_{\mathsf{T}}(\kappa_\pi(x_1), \ldots, \kappa_\pi(x_l)) \sim 0$ is true. As $|\kappa_{\pi'}(x) - \kappa_\pi(x)| \leq \delta$ for all freeze variables x occurring in ϕ, by the definition of relaxation, we have that

1. $f_{\mathsf{T}}(\kappa_\pi(x_1), \ldots, \kappa_\pi(x_l)) + K_{\mathsf{T}}(\delta) \sim 0$ if $\sim \in \{>, \geq\}$; and
2. $f_{\mathsf{T}}(\kappa_\pi(x_1), \ldots, \kappa_\pi(x_l)) - K_{\mathsf{T}}(\delta) \sim 0$ if $\sim \in \{<, \leq\}$.

This implies that $\kappa_{\pi'}$ is also a witness to the satisfaction of $\mathsf{rx}^\delta_{I_{\pi,\pi'}}(\phi)$ by π'. Thus, $\pi' \models \mathsf{rx}^\delta_{I_{\pi,\pi'}}(\phi)$. □

Definition 6 (δ-relaxation of TLTL$(\mathcal{F}_{\mathsf{T}}, \mathcal{F}_{\mathsf{S}})$ formulae). *Let ϕ be a* TLTL$(\mathcal{F}_{\mathsf{T}}, \mathcal{F}_{\mathsf{S}})$ *formula in which negations appear only on the prepositional symbols . The δ relaxation of ϕ (for $\delta \geq 0$), denoted $\mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi)$ is defined as follows, where $I_{\mathcal{F}_{\mathsf{T}}}$, a closed subset of reals$_+$, is the domain of the variables in V_{T}; and $\mathbf{I}_{V_{\mathsf{S}}}$ is a mapping from V_{S} to closed intervals of $\mathbb{R}$ such that $\mathbf{I}_{V_{\mathsf{S}}}(z)$ denotes the domain of z.*

$$\mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\text{TRUE}) = \text{TRUE}; \qquad \mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\text{FALSE}) = \text{FALSE};$$

$$\mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi_1 \wedge \phi_2) = \mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi_1) \wedge \mathsf{rx}_\delta(\phi_2);$$

$$\mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi_1 \vee \phi_2) = \mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi_1) \vee \mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi_2);$$

$$\mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(x.\psi) = x.\, \mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\psi);$$

$$\mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi_1 \,\mathcal{U}\, \phi_2) = \mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi_1) \,\mathcal{U}\, \mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi_2);$$

$$\mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi_1 \,\mathcal{W}\, \phi_2) = \mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi_1) \,\mathcal{W}\, \mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(\phi_2)$$

$$\mathsf{rx}^\delta_{I_{\mathcal{F}_{\mathsf{T}}}, \mathbf{I}_{V_{\mathsf{S}}}}(f_U(z_1, \ldots, z_l)) \sim 0) = \begin{cases} f_U(z_1, \ldots, z_l) + K_{f_U}(\delta) \sim 0 & \text{if } \sim \in \{>, \geq\}; \\ f_U(z_1, \ldots, z_l) - K_{f_U}(\delta) \sim 0 & \text{if } \sim \in \{<, \leq\}; \end{cases}$$

where $U \in \{\mathsf{T}, \mathsf{S}\}$ with K_{f_U} being as in Definition 4; and $K_{f_{\mathsf{S}}} : \left[0, \max_{z \in V_{\mathsf{S}}} |\max \mathbf{I}_{V_{\mathsf{S}}}(z) - \min \mathbf{I}_{V_{\mathsf{S}}}(z)|\right] \mapsto \mathbb{R}_+$ is a function such that:

$$K_{f_{\mathsf{S}}}(\delta) = \sup_{\substack{z_i \in \mathbf{I}_{V_{\mathsf{S}}}(z_i);\ z'_i \in \mathbf{I}_{V_{\mathsf{S}}}(z'_i) \\ \text{for all } i}} \left\{ \left| \begin{array}{c} f_{\mathsf{S}}(z_1, \ldots, z_l) \\ - \\ f_{\mathsf{S}}(z'_1, \ldots, z'_l) \end{array} \right| \text{ s.t. } |z_i - z'_i| \leq \delta \text{ for all } i \right\}$$

□

The functions $K_{f_{\mathsf{S}}}(\delta)$ define the maximal change in the value of f_{S} that can occur when the input variables can vary by δ. The role of $\mathbf{I}_{V_{\mathsf{S}}}$ in the above definition is to restrict the domain of the signal variables in order to obtain the

least possible bounds relaxation bounds on the signal constraints; as was done in Definition 4 for the freeze variables.

Proposition 2. *The function* $\mathsf{rx}^{\delta}_{I_{\mathcal{F}_\mathsf{T}},\mathbf{I}_{V_\mathsf{S}}}$ *is a relaxation on* $\mathrm{TLTL}(\mathcal{F}_\mathsf{T},\mathcal{F}_\mathsf{S})$ *formulae,* i.e. *if a timed* $\mathbb{R}^n$*-valued trace* $\pi \models \phi$ *for a* $\mathrm{TLTL}(\mathcal{F}_\mathsf{T},\mathcal{F}_\mathsf{S})$ *formula* ϕ*, then* $\pi \models \mathsf{rx}^{\delta}_{I_{\mathcal{F}_\mathsf{T}},\mathbf{I}_{V_\mathsf{S}}}(\phi)$.

Proof. The proof is similar to the proof of Proposition 1. □

Proof of Theorem 3. The proof use the result for the propositional case, Theorem 2. We construct the propositions p_{f_S} defined to be $\mathsf{rx}^{\delta}_{I_{\mathcal{F}_\mathsf{T}},\mathbf{I}_{V_\mathsf{S}}}(f_\mathsf{S}(\overline{y})) \sim 0)$ for the constraints over V_S in the formula ϕ; and define the $\mathrm{TLTL}(\mathcal{F}_\mathsf{T})$ formula $\phi_\mathcal{P}$ as that obtained from ϕ by syntactically replacing each constraint $f_\mathsf{S}(\overline{y}) \sim 0$ in ϕ by p_{f_S}. Let $\mathcal{P}_\mathsf{S}$ denote all such predicates for ϕ. We obtain the timed $\mathcal{P}_\mathsf{S}$ propositional traces $\pi_{P_\mathsf{S}}, \pi'_{P_\mathsf{S}}$ from π, π' by mapping to propositions. By the definition of the skorokhod distance, the distance between $\pi_{\mathcal{P}_\mathsf{S}}$ and $\pi'_{\mathcal{P}_\mathsf{S}}$ is less than δ. By Theorem 2, $\pi'_{\mathcal{P}_\mathsf{S}} \models \phi_\mathcal{P}$. This implies $\pi' \models \mathsf{rx}^{\delta}_{I_{\mathcal{F}_\mathsf{T}},\mathbf{I}_{V_\mathsf{S}}}(\phi)$. □

B. Details on Case Studies

LQR-based pitch controller. The aircraft pitch controller system has 3 state variables, and the state vector $\mathbf{x} = [\alpha\ q\ \theta]$, 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). In deriving the control law, the designers use the state feedback law to substitute $\delta = \theta_{des} - K\mathbf{x}$, where θ_{des} is the desired pitch angle. 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:

$$A = \begin{bmatrix} -0.313 & 56.7 & 0 \\ -0.0139 & -0.426 & 0 \\ 0 & 56.7 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0.232 \\ 0.0203 \\ 0 \end{bmatrix}$$

$$K = [-0.6435 \quad 169.6950 \quad 7.0711]$$

Air-Fuel Ratio Controller. The Air-Fuel (A/F) ratio control systems that we consider are simplified versions of industrial-scale models. Both versions have 2 exogenous inputs, and 4 continuous states. 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 continuous states, we assume that 2 of these states are from the plant model (that encapsulates physical processes within the engine), while 2 states belong to the controller. The plant states p and

λ denote intake manifold pressure and the A/F ratio respectively. The controller states p_e denotes the estimated manifold pressure (with the use of an observer) used in the feed-forward control, and the state 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 [19, 20].