Temporal Logic Verification Using Simulation

Georgios E. Fainekos[1], Antoine Girard[2], and George J. Pappas[3]

[1] Department of Computer and Information Science, Univ. of Pennsylvania, USA
fainekos@cis.upenn.edu
[2] VERIMAG, 2 avenue de Vignate, 38610 Gières, France
Antoine.Girard@imag.fr
[3] Department of Electrical and Systems Engineering, Univ. of Pennsylvania, USA
pappasg@ee.upenn.edu

Abstract. In this paper, we consider a novel approach to the temporal logic verification problem of continuous dynamical systems. Our methodology has the distinctive feature that enables the verification of the temporal properties of a continuous system by verifying only a finite number of its (simulated) trajectories. The proposed framework comprises two main ideas. First, we take advantage of the fact that in metric spaces we can quantify how close are two different states. Based on that, we define robust, multi-valued semantics for MTL (and LTL) formulas. These capture not only the usual Boolean satisfiability of the formula, but also topological information regarding the distance from unsatisfiability. Second, we use the recently developed notion of bisimulation functions to infer the behavior of a set of trajectories that lie in the neighborhood of the simulated one. If the latter set of trajectories is bounded by the tube of robustness, then we can infer that all the trajectories in the neighborhood of the simulated one satisfy the same temporal specification as the simulated trajectory. The interesting and promising feature of our approach is that the more robust the system is with respect to the temporal logic specification, the less is the number of simulations that are required in order to verify the system.

1 Introduction

Software and hardware design has tremendously benefited from advances in algorithmic verification. Model checking [1] is now a widely used technology in various industrial settings. Thanks to the rapidly growing area of embedded systems with real-time specifications, a similar growth is also being experienced in the area of real-time systems [2]. As the complexity of the physical systems increases and captures continuous or hybrid systems, the verification problems quickly become hard, if not undecidable.

For the verification of hybrid systems, a variety of methods have been proposed [3,4,5,6,7,8] (not an inclusive list). The common characteristic of all these approaches is that they apply to either continuous systems with simple dynamics, or they are computationally expensive and, thus, they can only be used for low dimensional systems (for promising high-dimensional results see [9,10]). Beyond the scope of these techniques, the analysis of complex systems still relies

E. Asarin and P. Bouyer (Eds.): FORMATS 2006, LNCS 4202, pp. 171–186, 2006.

heavily on simulation-based methods for monitoring [11]. Along these lines several authors have proposed simulation techniques that can provide guarantees for uniform coverage [12,13] or even completeness results [14].

This paper develops a simulation-based method for verifying temporal properties of complex continuous systems. In particular, given a continuous dynamical system, a set of initial conditions, a bounded time horizon, and a temporal logic specification expressed in Metric or Linear Temporal Logic [15], we develop a simulation-based algorithm that verifies whether all the system trajectories satisfy the desired temporal property. To achieve this, we build upon two recent notions : a definition of *robust satisfaction* for Metric Temporal Logic (MTL) specifications [16] and the notion of *bisimulation functions* [17]. The definition of robust satisfaction of an MTL specification is meaningful only when state sequences evolve in metric spaces, a very natural assumption for continuous systems. Our proposed robust semantics capture bounds on the magnitude of the state perturbations that can be tolerated without altering the Boolean truth value of the MTL or LTL property. Bisimulation functions, on the other hand, quantify the distance between two approximately bisimilar states and the trajectories initiating from them. Using a bisimulation function we can define a neighborhood of trajectories around a nominal one which have approximately the same behavior as the nominal trajectory. If this neighborhood of the simulated trajectory is contained in the tube of trajectories, which robustly satisfy the specification, then we can safely infer that the neighborhood of trajectories also satisfies the specification.

Based on this observation, we develop an algorithm that, first, samples points in the set of initial conditions of the system using guidance from the bisimulation function. Starting from this set of points, we simulate the system for a bounded horizon. For each of these trajectories we compute an under-approximation of its robustness degree. If the robustness degree bounds the distance computed by the bisimulation function then we are done, otherwise we repeat the procedure. The novelty in our framework is that the number of simulations, which are required for the verification of the system, decreases inversely to the robustness of the system with respect to the temporal property.

Finally, we would like to point out that in the past several authors have also studied the robustness of real time specifications with respect to timed or dense time traces of real time systems [18,19,20], but the robustness is considered with respect to the timing constraints, not state perturbations. The work which is the closest in spirit to this paper appears in [21] where the authors give quantitative semantics to the branching-time logic CTL (called Discounted CTL) in order to achieve robustness with respect to model perturbations.

2 Problem Formulation

Let $\mathbb{R}$ be the set of the real numbers, $\mathbb{Q}$ the set of the rational numbers and $\mathbb{N}$ the set of the natural numbers. We denote the extended real number line by

$\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$. In addition, $\mathbb{R}_{\geq 0}$ denotes the subset of the reals whose elements are greater or equal to zero. Finally, $\mathcal{P}(C)$ denotes the powerset of a set C.

2.1 Continuous Time Dynamical Systems as Timed State Sequences

In this paper, we focus on the verification of temporal properties of continuous time dynamical systems.

Definition 1 (Continuous Time Dynamical System). *A continuous-time dynamical system is defined by a tuple $\Sigma = (N, P, f, g, I, AP, \mathcal{O})$ where: N and P are positive integers which respectively denote the dimension of the state-space and of the observation-space, $f : \mathbb{R}^N \to \mathbb{R}^N$ and $g : \mathbb{R}^N \to \mathbb{R}^P$ are continuous maps, I is a compact subset of $\mathbb{R}^N$ which denotes the set of initial states, AP is a set of atomic propositions and $\mathcal{O} : AP \to \mathcal{P}(\mathbb{R}^P)$ is a predicate mapping.*

A *trajectory* of the continuous-time dynamical system Σ is a pair of functions $(x(t), y(t))$ such that $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^N$ and $y : \mathbb{R}_{\geq 0} \to \mathbb{R}^P$ satisfy $x(0) \in I$ and $\forall t \in \mathbb{R}_{\geq 0}$

$$\dot{x}(t) = f(x(t)) \text{ and } y(t) = g(x(t)) \tag{1}$$

Here, $\dot{x}$ denotes the first order derivative of the function x. We make the standard assumption on f so that for a given initial state $x_0 \in I$, the system Σ has a unique trajectory. Hence, given $x(0)$ the system is deterministic. In order to specify the properties of interest for the system Σ in any temporal logic, we must define a set of regions in its observation space. If AP is a finite set of atomic propositions, then the *predicate mapping* $\mathcal{O} : AP \to \mathcal{P}(\mathbb{R}^P)$ is a set valued function that assigns to each atomic proposition $\pi \in AP$ a set of states $\mathcal{O}(\pi) \subseteq \mathbb{R}^P$.

Beyond certain classes of continuous dynamical systems such as linear systems, system Σ does not always have an analytical solution. Typically, the behavior of such systems can be explored using numerical simulation [22]. Numerical simulation methods approximate the differential equations of the system Σ by algebraic equations which depend on the size of the integration (time) step. Furthermore, such simulations can only be of finite duration. Therefore, we can model such computations by finite timed state sequences.

Definition 2 (TSS). *A timed state sequence $\mathcal{T}$ in a space Q is a tuple $(\sigma, \tau, \mathcal{O})$ where for some $n \in \mathbb{N}$: $\sigma = \sigma_0, \sigma_1, \ldots, \sigma_n$ is a sequence of states, $\tau = \tau_0, \tau_1, \ldots, \tau_n$ is a sequence of time stamps and $\mathcal{O} : AP \to \mathcal{P}(Q)$ is a predicate mapping. The following conditions must be satisfied by $\mathcal{T}$: (i) for all $i \in \{0, 1, \ldots, n\}$ we have $\sigma_i \in Q$ and $\tau_i \in \mathbb{R}_{\geq 0}$ and (ii) τ is a strictly monotonically increasing sequence.*

By convention, we set $\tau_0 = 0$ (in the Metric Temporal Logic we care only about relative time). We define $\sigma\uparrow_i$ to be the suffix of a sequence, i.e. $\sigma\uparrow_i = \sigma_i, \sigma_{i+1}, \ldots, \sigma_n$. When the same operator $\uparrow_i$ is applied to the sequence τ, it is defined as $\tau\uparrow_i = 0, \tau_{i+1} - \tau_i, \ldots, \tau_n - \tau_i$. The length of $\sigma = \sigma_0, \sigma_1, \ldots, \sigma_n$ is defined to be $|\sigma| = n + 1$. For convenience, we let $|\mathcal{T}| = |\tau| = |\sigma|$ and $\mathcal{T}\uparrow_i = (\sigma\uparrow_i, \tau\uparrow_i, \mathcal{O})$.

We define TS to be the set of all possible finite timed state sequences in the space $\mathbb{R}^P$, that is $\mathrm{TS} = \{(\sigma, \tau, \mathcal{O}) \mid n \in \mathbb{N}_{>0}, \sigma \in (\mathbb{R}^P)^n, \tau \in \mathbb{R}^n_{\geq 0}$ such that $\tau_i < \tau_{i+1}$ for $i < n$ and $\mathcal{O} : AP \to \mathcal{P}(\mathbb{R}^P)\}$. Note that by definition we do not consider empty timed state sequences and that essentially the sequence σ is isomorphic to a point in the product space $(\mathbb{R}^P)^{|\sigma|}$. In addition, given a timed state sequence $\mathcal{T} = (\sigma, \tau, \mathcal{O})$, then $\mathrm{TS}_{\mathcal{T}}$ is the set of all timed state sequences with the same predicate mapping $\mathcal{O}$ and the same sequence of time stamps τ as $\mathcal{T}$, i.e. $\mathrm{TS}_{\mathcal{T}} = \{(\sigma', \tau', \mathcal{O}') \in \mathrm{TS} \mid \tau' = \tau, \mathcal{O}' = \mathcal{O}\}$.

Now, given a sequence of integration steps (which is equivalent to a sequence of time stamps τ) for the numerical simulation of the system Σ, we can model the resulting discrete trajectory as a timed state sequence, which we refer to as a trace.

Definition 3 (Trace). *Given a sequence of time stamps τ, a trace of a continuous dynamical system Σ is a timed state sequence $\mathcal{T} = (\sigma, \tau, \mathcal{O})$ such that there exists a trajectory (x, y) of Σ satisfying $\sigma_i = y(\tau_i) = g(x(\tau_i))$ for all $i = 0, 1, \ldots, |\tau| - 1$. The set of traces of Σ associated with the sequence of time stamps τ is denoted by $\mathcal{L}_\tau(\Sigma)$.*

We should point out that in this paper, we essentially consider the trace to be sampled from the continuous solution of the system Σ. In numerical methods for the integration of differential equations, though, there exists a quantifiable and bounded error between the continuous solution of the equations (1) and the result of the numerical simulation, which can be driven arbitrarily close to zero [22]. Therefore, we can safely ignore this issue for now in order to facilitate the presentation of the contributions in the current paper.

2.2 Metric Temporal Logic over Finite Timed State Sequences

We employ the Metric Temporal Logic (MTL) [15] in order to formally characterize the desired behavior of the system Σ. In MTL, the syntax of the logic is extended to include timing constraints on the usual temporal operators of the Linear Temporal Logic (LTL). Using LTL specifications we can check qualitative timing properties, while with MTL specifications quantitative timing properties. Recently, it was shown by Ouaknine and Worrell [23] that MTL is decidable over finite timed state sequences. In this section, we review the basics of MTL with point-based semantics over finite timed state sequences.

Definition 4 (Syntax of MTL). *An MTL formula ϕ is inductively defined according to the grammar*

$$\phi ::= \top \mid \pi \mid \neg\phi_1 \mid \phi_1 \vee \phi_2 \mid \phi_1 \mathcal{U}_{\mathcal{I}} \phi_2$$

where $\pi \in AP$, $\top$ is the symbol for the boolean constant true and $\mathcal{I}$ is an interval of $\mathbb{R}_{\geq 0}$ with rational endpoints.

The set of all well formed MTL formulas is denoted by Φ_{MTL}. Even though we can derive the constant true ($\top$) from the law of excluded middle ($\top = \pi \vee \neg\pi$), we chose to add it in the syntax of MTL for reasons that will be clear in Sect. 3. The constant *false* is denoted by $\bot = \neg\top$. We can also derive additional temporal operators such as *release* $\phi_1 \mathcal{R}_{\mathcal{I}} \phi_2 = \neg((\neg\phi_1)\mathcal{U}_{\mathcal{I}}\neg\phi_2)$ (which is the dual of the until operator), *eventually* $\Diamond_{\mathcal{I}}\phi = \top\,\mathcal{U}_{\mathcal{I}}\phi$ and *always* $\Box_{\mathcal{I}}\phi = \bot\,\mathcal{R}_{\mathcal{I}}\phi$.

The subscript $\mathcal{I}$ imposes timing constraints on the temporal operators. The interval $\mathcal{I}$ can be open, half-open or closed, bounded or unbounded, or even a singleton. For any $t \in \mathbb{Q}$, we define $\mathcal{I} + t = \{t' + t \mid t' \in \mathcal{I}\}$. In the case where $\mathcal{I} = [0, +\infty)$, we remove the subscript $\mathcal{I}$ from the temporal operators, i.e. we just write $\mathcal{U}$, $\mathcal{R}$, $\Diamond$ and $\Box$. When all the subscripts of the temporal operators are of the form $[0, +\infty)$, then the MTL formula ϕ reduces to an LTL formula and we can ignore the time stamps.

Metric Temporal Logic (MTL) formulas are interpreted over timed state sequences $\mathcal{T}$ with $|\mathcal{T}| > 0$. In this paper, we denote formula satisfiability using a membership function $\langle\!\langle\phi\rangle\!\rangle : \text{TS} \to \{\bot, \top\}$ instead of the usual notation $\mathcal{T} \models \phi$. We say that a timed state sequence $\mathcal{T}$ satisfies the formula ϕ when $\langle\!\langle\phi\rangle\!\rangle(\mathcal{T}) = \top$. In this case, $\mathcal{T}$ is a *model* of ϕ. The set of all models of ϕ is denoted by $\mathcal{L}(\phi)$, i.e. $\mathcal{L}(\phi) = \{\mathcal{T} \in \text{TS} \mid \langle\!\langle\phi\rangle\!\rangle(\mathcal{T}) = \top\}$.

Definition 5 (Semantics of MTL). *Let $\mathcal{T} = (\sigma, \tau, \mathcal{O}) \in TS$, $\pi \in AP$, $i, j \in \mathbb{N}$ and $\psi, \phi_1, \phi_2 \in \Phi_{\text{MTL}}$, then the semantics of any MTL formula ϕ are defined recursively as*

$$
\begin{aligned}
\langle\!\langle\top\rangle\!\rangle(\mathcal{T}) &:= \top \\
\langle\!\langle\pi\rangle\!\rangle(\mathcal{T}) &:= \sigma_0 \in \mathcal{O}(\pi) \\
\langle\!\langle\neg\psi\rangle\!\rangle(\mathcal{T}) &:= \neg\langle\!\langle\psi\rangle\!\rangle(\mathcal{T}) \\
\langle\!\langle\phi_1 \vee \phi_2\rangle\!\rangle(\mathcal{T}) &:= \langle\!\langle\phi_1\rangle\!\rangle(\mathcal{T}) \vee \langle\!\langle\phi_2\rangle\!\rangle(\mathcal{T}) \\
\langle\!\langle\phi_1\,\mathcal{U}_{\mathcal{I}}\phi_2\rangle\!\rangle(\mathcal{T}) &:= \bigvee\nolimits_{i=0}^{|\mathcal{T}|-1}\big((\tau_i \in \mathcal{I}) \wedge \langle\!\langle\phi_2\rangle\!\rangle(\mathcal{T}{\uparrow_i}) \wedge \bigwedge\nolimits_{j=0}^{i-1}\langle\!\langle\phi_1\rangle\!\rangle(\mathcal{T}{\uparrow_j})\big)
\end{aligned}
$$

2.3 Problem Statement

Now that we have presented all the necessary mathematical objects we can formally state the verification problem that we answer in this paper.

Problem 6. Given an MTL formula ϕ, a continuous dynamical system Σ and a sequence of time stamps τ, verify that $\mathcal{L}_\tau(\Sigma) \subseteq \mathcal{L}(\phi)$. In other words, verify that all the traces $\mathcal{T}$ of Σ satisfy the specification ϕ.

The difficulty in solving Problem 6 is that in metric spaces there exists an infinite number of traces $\mathcal{T} = (\sigma, \tau, \mathcal{O})$ of Σ. Thus, the verification of Σ cannot be done by exhaustive simulation. In the following, we show that using the robust semantics of MTL (Sect. 3) and the notion of bisimulation function [17], the verification of Σ is possible by using only a finite number of simulations.

Example 7. In order to motivate the rest of the discussion, we present as an example the verification problem of a transmission line [10]. The goal is to check that the transient behavior of a long transmission line is acceptable both in terms of overshoot and of response time. Figure 1 shows a model of the transmission line, which consists of a number of RLC components (R: resistor, L: inductor and C: capacitor) modeling segments of the line. The left side is the sending end and the right side is the receiving end of the transmission line.

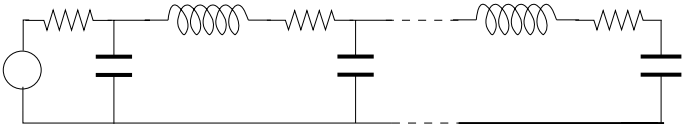

Fig. 1. RLC model of a transmission line

The dynamics of the system are given by the linear dynamical system

$$\dot{x}(t) = Ax(t) + bU_{in}(t) \text{ and } U_{out}(t) = Cx(t)$$

where $x(t) \in \mathbb{R}^N$ is the state vector containing the voltage of the capacitors and the current of the inductors and $U_{in}(t) \in \mathbb{R}$ is the voltage at the sending end. The output of the system is the voltage $U_{out}(t) \in \mathbb{R}$ at the receiving end. Here, A, b and c are matrices of appropriate dimensions. Initially, $U_{in}(0) \in [-0.2, 0.2]$ and the system is at its steady state $x(0) = -A^{-1}bU_{in}(0)$. Then, at time $t = 0$ the input is set to the value $U_{in}(t) = 1$. We use an 81st order RLC model of the transmission line (i.e. $N = 81$). An example of a trace is shown in Fig. 2.

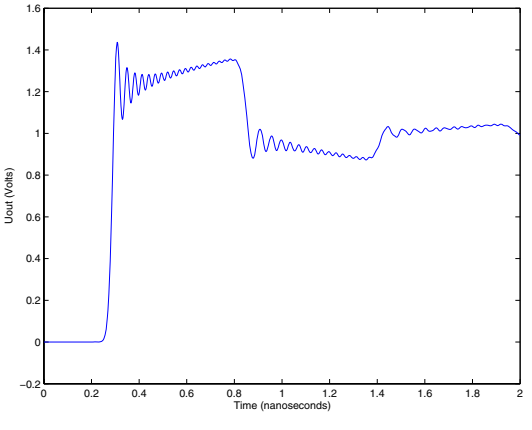

Fig. 2. An example trace of the RLC model of the transmission line

The goal of the verification is double. We want to check that the voltage at the receiving end stabilizes between 0.8 and 1.2 Volts within T nano-seconds (response time) and that its amplitude always remains bounded by θ Volts (overshoot) where $T \in [0, 2]$ and $\theta \geq 0$ are design parameters. The specification is expressed as the MTL property:

$$\phi = \Box \pi_1 \wedge \Diamond_{[0,T]} \Box \pi_2$$

where the predicates are mapped as follows: $\mathcal{O}(\pi_1) = [-\theta, \theta]$ and $\mathcal{O}(\pi_2) = [0.8, 1.2]$. We consider a time frame of 2 nanoseconds. The sequence of time stamps τ is uniformly generated with a time step of $\Delta t = 0.02$ nanoseconds.

3 Robust Satisfaction of MTL Specifications

In this section, we define what it means for a timed state sequence to satisfy a Metric Temporal Logic specification *robustly*. Our definition of robustness is built upon the fact that in metric spaces we can quantify how far apart are two points of the space.

3.1 Distance in Metric Spaces

Let (Q, d) be a metric space, that is a set Q with a metric d which gives the topology of Q. Given two points q_1, q_2 of Q, the number $d(q_1, q_2)$ is called the *distance* between q_1 and q_2 in the metric d. Using the metric d, we can define the distance of a point $q \in Q$ from a subset of $R \subseteq Q$.

Definition 8 (Distance, Signed Distance). *Let $q \in Q$ be a point, $R \subseteq Q$ be a set and d be a metric. Then we define the*

- distance *from q to R to be* $\mathbf{dist}_d(q, R) := \inf\{d(q, q') \mid q' \in R\}$
- signed distance *from q to R to be*

$$\mathbf{Dist}_d(q, R) := \begin{cases} -\mathbf{dist}_d(q, R) & \text{if } q \notin R \\ \mathbf{dist}_d(q, Q \backslash R) & \text{if } q \in R \end{cases}$$

We should point out that we use the extended definition of supremum and infimum, where $\sup \emptyset = -\infty$ and $\inf \emptyset = +\infty$. Also of importance is the notion of an open ball of radius ε centered at a point $q \in Q$.

Definition 9 (ε-Ball). *Given a metric d, a radius $\varepsilon \in \overline{\mathbb{R}}_{>0}$ and a point $q \in Q$, the open ε-ball centered at q is defined as $B_d(q, \varepsilon) = \{q' \in Q \mid d(q, q') < \varepsilon\}$.*

If the distance ($\mathbf{dist}_d$) of a point q from a set R is $\varepsilon > 0$, then $B_d(q, \varepsilon) \cap R = \emptyset$. And similarly, if $\mathbf{dist}_d(q, Q \backslash R) = \varepsilon > 0$, then $B_d(q, \varepsilon) \subseteq R$.

3.2 Defining Robust Semantics for the Metric Temporal Logic

In $\mathbb{R}^P$, we can quantify how close are two different observations $y_1, y_2 \in \mathbb{R}^P$ by using the metric $d(y_1, y_2) = \|y_1 - y_2\| = \sqrt{(y_1 - y_2)^T (y_1 - y_2)}$. Let $\mathcal{T} = (\sigma, \tau, \mathcal{O})$ be a timed state sequence and $(\sigma', \tau, \mathcal{O}) \in \mathrm{TS}_{\mathcal{T}}$, then

$$\rho(\sigma, \sigma') = \max\{\|\sigma_0 - \sigma'_0\|, \|\sigma_1 - \sigma'_1\|, \ldots, \|\sigma_{|\tau|-1} - \sigma'_{|\tau|-1}\|\} \tag{2}$$

is a metric on the set $(\mathbb{R}^P)^{|\mathcal{T}|}$, which is well defined since $|\mathcal{T}|$ is finite. Now that the space of state sequences is equipped with a metric, we can define a tube around a timed state sequence $\mathcal{T}$. Given an $\varepsilon > 0$, then

$$\mathrm{TS}_{\mathcal{T}}^{\varepsilon} = \{(\sigma', \tau, \mathcal{O}) \in \mathrm{TS}_{\mathcal{T}} \mid \sigma' \in B_\rho(\sigma, \varepsilon)\}$$

is the set of all timed state sequences that remain ε-close to $\mathcal{T}$.

Informally, we define the degree of robustness that a timed state sequence $\mathcal{T}$ satisfies an MTL formula ϕ to be a number $\varepsilon \in \overline{\mathbb{R}}$. Intuitively, a positive ε means that the formula ϕ is satisfiable and, moreover, that all the other timed state sequences that remain ε-close to the nominal one also satisfy ϕ. Accordingly, if ε is negative, then $\mathcal{T}$ does not satisfy ϕ and all the other timed state sequences that remain within the open tube of radius $|\varepsilon|$ also do not satisfy ϕ.

Definition 10 (Degree of Robustness). *Let $\phi \in \Phi_{\mathrm{MTL}}$, $\mathcal{T} = (\sigma, \tau, \mathcal{O}) \in TS$ and ρ be the metric (2). Define $P_{\mathcal{T}}^{\phi} := \{\sigma' \mid (\sigma', \tau, \mathcal{O}) \in TS_{\mathcal{T}} \cap \mathcal{L}(\phi)\}$, then the robustness degree $\varepsilon \in \overline{\mathbb{R}}$ of $\mathcal{T}$ with respect to ϕ is defined as $\varepsilon := \mathbf{Dist}_\rho(\sigma, P_{\mathcal{T}}^{\phi})$.*

The following proposition is derived directly from the definitions. It states that all the timed state sequences $\mathcal{S}$, which have distance from $\mathcal{T}$ less than robustness degree of $\mathcal{T}$, satisfy the same specification ϕ as $\mathcal{T}$.

Proposition 11. *Let $\phi \in \Phi_{\mathrm{MTL}}$, $\mathcal{T} = (\sigma, \tau, \mathcal{O}) \in TS$ and $\varepsilon = \mathbf{Dist}_\rho(\sigma, P_{\mathcal{T}}^{\phi})$. If $|\varepsilon| > 0$, then $\langle\!\langle \phi \rangle\!\rangle(\mathcal{S}) = \langle\!\langle \phi \rangle\!\rangle(\mathcal{T})$ for all $\mathcal{S} \in TS_{\mathcal{T}}^{|\varepsilon|}$.*

Remark 12. If $\varepsilon = 0$, then the truth value of ϕ with respect to $\mathcal{T}$ is not robust, i.e. any small perturbation of a critical state in the timed state sequence can change the satisfiability of the formula with respect to $\mathcal{T}$.

Theoretically, the set $P_{\mathcal{T}}^{\phi}$ can be computed since we have a finite number of atomic propositions, a finite trace and a known in advance sequence of time stamps. Implementation wise, though, the construction of the set $P_{\mathcal{T}}^{\phi}$ and the computation of the distance $\mathbf{Dist}_\rho(\sigma, P_{\mathcal{T}}^{\phi})$ are computationally expensive, if not infeasible (for a discussion see [16]). Therefore in this section, we develop an algorithm that computes a conservative approximation of the robustness degree ε by directly operating on the timed state sequence while avoiding set operations. As is usually the case in trade-offs, we gain computational efficiency at the expense of accuracy.

Similar to [21], we propose multi-valued semantics for the Metric Temporal Logic where the valuation function on the atomic propositions takes values over

the totally ordered set $\mathfrak{R} = (\overline{\mathbb{R}}, \leq)$ according to the metric d operating on the state space $\mathbb{R}^P$ of the timed state sequence $\mathcal{T}$. For this purpose, we let the valuation function be the signed distance from the current point in the trace y to a set $\mathcal{O}(\pi)$. Intuitively, this distance represents how robustly is a point y within the set $\mathcal{O}(\pi)$. If this metric is zero, then even the smallest perturbation of the point can drive it inside or outside the set $\mathcal{O}(\pi)$, dramatically affecting membership.

We define the binary operators $\sqcup : \overline{\mathbb{R}} \times \overline{\mathbb{R}} \to \overline{\mathbb{R}}$ and $\sqcap : \overline{\mathbb{R}} \times \overline{\mathbb{R}} \to \overline{\mathbb{R}}$ using the maximum and minimum functions as $\alpha \sqcup \beta := \max\{\alpha, \beta\}$ and $\alpha \sqcap \beta := \min\{\alpha, \beta\}$. Also, for some $R \subseteq \overline{\mathbb{R}}$ we extend the above definitions as follows: $\bigsqcup R := \sup R$ and $\bigsqcap R := \inf R$. Recall that $\bigsqcup \overline{\mathbb{R}} = +\infty$ and $\bigsqcap \overline{\mathbb{R}} = -\infty$ and that any subset of $\overline{\mathbb{R}}$ has a supremum and infimum. Finally, because $\mathfrak{R}$ is a totally ordered set, it is distributive, i.e. for all $\alpha, \beta, \gamma \in \overline{\mathbb{R}}$ it is $\alpha \sqcap (\beta \sqcup \gamma) = (\alpha \sqcap \beta) \sqcup (\alpha \sqcap \gamma)$.

For the purposes of the following discussion, we use the notation $[\![\phi]\!](\mathcal{T})$ to denote the approximation to the degree of robustness with which the structure $\mathcal{T}$ satisfies the specification ϕ (formally $[\![\phi]\!] : \mathrm{TS} \to \overline{\mathbb{R}}$).

Definition 13 (Robust Semantics of MTL). *For $\phi \in \Phi_{\mathrm{MTL}}$ and $\mathcal{T} = (\sigma, \tau, \mathcal{O}) \in TS$, the robust semantics of ϕ with respect to $\mathcal{T}$ are defined as (let $\pi \in AP$ and $\psi, \phi_1, \phi_2 \in \Phi_{\mathrm{MTL}}$)*

$$
\begin{aligned}
[\![\top]\!](\mathcal{T}) &:= +\infty \\
[\![\pi]\!](\mathcal{T}) &:= \mathbf{Dist}_d(\sigma_0, \mathcal{O}(\pi)) \\
[\![\neg\psi]\!](\mathcal{T}) &:= -[\![\psi]\!](\mathcal{T}) \\
[\![\phi_1 \vee \phi_2]\!](\mathcal{T}) &:= [\![\phi_1]\!](\mathcal{T}) \sqcup [\![\phi_2]\!](\mathcal{T}) \\
[\![\phi_1 \,\mathcal{U}_{\mathcal{I}} \phi_2]\!](\mathcal{T}) &:= \bigsqcup_{i=0}^{|\mathcal{T}|-1} \big([\![\tau_i \in \mathcal{I}]\!](\mathcal{T}) \sqcap [\![\phi_2]\!](\mathcal{T}\!\uparrow_i) \sqcap \bigsqcap_{j=0}^{i-1} [\![\phi_1]\!](\mathcal{T}\!\uparrow_j) \big)
\end{aligned}
$$

where the unary operator $(-)$ is defined to be the negation over the reals.

Remark 14. It is easy to verify that the semantics of the negation operator give us all the usual nice properties such as the *De Morgan laws*: $a \sqcup b = -(-a \sqcap -b)$ and $a \sqcap b = -(-a \sqcup -b)$, *involution*: $-(-a) = a$ and *antisymmetry*: $a \leq b$ iff $-a \geq -b$ for $a, b \in \overline{\mathbb{R}}$.

The following theorem states that robustness parameter that we compute using the robust semantics of MTL is an under-approximation of the actual degree of robustness (for the proof see the technical report [16]).

Theorem 15. *Let $\phi \in \Phi_{\mathrm{MTL}}$ and $\mathcal{T} \in TS$, then $|[\![\phi]\!](\mathcal{T})| \leq |\mathbf{Dist}_\rho(\sigma, P^\phi_{\mathcal{T}})|$. Moreover, if $[\![\phi]\!](\mathcal{T}) = \varepsilon \neq 0$, then for all $\mathcal{S} \in TS^{|\varepsilon|}_{\mathcal{T}}$ it is $\langle\!\langle \phi \rangle\!\rangle(\mathcal{S}) = \langle\!\langle \phi \rangle\!\rangle(\mathcal{T})$.*

Based on the robust semantics of MTL, we can derive a tableau formulation of the until temporal operator which is the basis for an on-the-fly monitoring algorithm. For the state of art monitoring algorithms for MTL using point-based semantics see [24] and the references therein. Using similar procedures, it is easy to derive an algorithm that returns the Boolean truth value of the formula and its robustness degree. Further details can be found in the technical report [16].

Remark 16. Consider the MTL fragment $\Phi_{\text{MTL}}(\wedge, \Box)$ where the only allowed operators are the conjunction and always. In this fragment, the negation ($\neg$) can appear only in front of atomic propositions. If $\phi \in \Phi_{\text{MTL}}(\wedge, \Box)$ and $\langle\!\langle \phi \rangle\!\rangle(\mathcal{T}) = \top$, then $[\![\phi]\!](\mathcal{T}) = \mathbf{Dist}_\rho(\sigma, P^\phi_{\mathcal{T}})$. For a discussion see [16].

4 Verification Using Robust Simulation

In this section, we show that Problem 6 can be solved in the framework of continuous-time dynamical systems. Our approach comprises three basic steps. First, we define a notion of neighborhood on the set of trajectories of the system Σ. This enables us to determine the sets of trajectories with approximately equivalent behaviors. Then, it is possible to verify that a property ϕ holds for all the traces of the dynamical system by simulating only a finite number of traces of the system Σ and evaluating their coefficients of robustness. A similar approach was proposed for the verification of safety properties in [14].

4.1 Bisimulation Function

The notion of bisimulation function has been introduced in [17] in the context of general non-deterministic metric transition systems. Intuitively, a bisimulation function evaluates how far are two states from being bisimilar: a bisimulation function bounds the distance between the observations associated with the states and is non-increasing during the evolution of the system. In the context of continuous-time dynamical systems considered in this paper, the formal definition of a bisimulation function is the following.

Definition 17. *A continuous function $V : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}_{\geq 0}$ is a bisimulation function for the dynamical system Σ if it satisfies the following properties*

1. *For all $x \in \mathbb{R}^N$, $V(x, x) = 0$*
2. *For all $x_1 \in \mathbb{R}^N$, $x_2 \in \mathbb{R}^N$, $V(x_1, x_2) \geq \|g(x_1) - g(x_2)\|$*
3. *For all $x_1 \in \mathbb{R}^N$, $x_2 \in \mathbb{R}^N$,*

$$\frac{\partial V}{\partial x_1}(x_1, x_2) \cdot f(x_1) + \frac{\partial V}{\partial x_2}(x_1, x_2) \cdot f(x_2) \leq 0.$$

Remark 18. Effective characterizations of bisimulation functions have been proposed for linear dynamical systems based on a set of linear matrix inequalities [25] and for nonlinear dynamical systems based on sum of squares programs [26]. Both characterizations can be interpreted in terms of convex optimization leading to efficient algorithms for the computation of bisimulation functions.

Theorem 19. *Let V be a bisimulation function, (x_1, y_1) and (x_2, y_2) be trajectories of Σ and $\mathcal{T}_1 = (\sigma^1, \tau, \mathcal{O}) \in \mathcal{L}_\tau(\Sigma)$ and $\mathcal{T}_2 = (\sigma^2, \tau, \mathcal{O}) \in \mathcal{L}_\tau(\Sigma)$ be the associated traces, then*

$$(\exists i \in \{1, 2\}. V(x_1(0), x_2(0)) < |[\![\phi]\!](\mathcal{T}_i)|) \implies (\langle\!\langle \phi \rangle\!\rangle(\mathcal{T}_1) = \langle\!\langle \phi \rangle\!\rangle(\mathcal{T}_2)) \quad (3)$$

Proof. From the third property of Definition 17, it follows that for all $t \in \mathbb{R}_{\geq 0}$,

$$\frac{dV(x_1(t), x_2(t))}{dt} = \frac{\partial V}{\partial x_1}(x_1(t), x_2(t)) \cdot f(x_1(t)) + \frac{\partial V}{\partial x_2}(x_1(t), x_2(t)) \cdot f(x_2(t)) \leq 0.$$

Then, from the second property of Definition 17, for all $t \in \mathbb{R}_{\geq 0}$,

$$\|y_1(t) - y_2(t)\| = \|g(x_1(t)) - g(x_2(t))\| \leq V(x_1(t), x_2(t)) \leq V(x_1(0), x_2(0)).$$

Therefore, $\forall i \in \{0, 1, \ldots, |\tau| - 1\}$, it is $\|y_1(\tau_i) - y_2(\tau_i)\| \leq V(x_1(0), x_2(0))$ or

$$\rho(\sigma^1, \sigma^2) \leq V(x_1(0), x_2(0)) \tag{4}$$

Without loss of generality assume that $V(x_1(0), x_2(0)) < |[\![\phi]\!](\mathcal{T}_1)|$ and let $\varepsilon' = V(x_1(0), x_2(0))$ and $\varepsilon = |[\![\phi]\!](\mathcal{T}_1)|$. Equation (4) implies that $\mathcal{T}_2$ belongs in the closure of $TS^{\varepsilon'}_{\mathcal{T}_1}$. But $TS^{\varepsilon}_{\mathcal{T}_1} \subset TS^{\varepsilon}_{\mathcal{T}_1}$ since $\varepsilon' < \varepsilon$. Therefore, $\mathcal{T}_2 \in TS^{\varepsilon}_{\mathcal{T}_1}$ and by applying Theorem 15 we can conclude that $\langle\!\langle \phi \rangle\!\rangle(\mathcal{T}_1) = \langle\!\langle \phi \rangle\!\rangle(\mathcal{T}_2)$. □

The previous result means that using the robust semantics of MTL and a bisimulation function, it is possible to infer the Boolean truth value of the MTL specification for an infinite number of traces. This property is exploited in the following section to verify all the traces of a system Σ using only a finite number of traces.

4.2 Sampling the Initial States

The challenge in developing a simulation-based verification algorithm is to sample the set of initial conditions in a way that ensures coverage. For this purpose, we define a discretization operator based on the bisimulation function.

Proposition 20. *Let V be a bisimulation function. For any compact set of initial conditions $I \subseteq \mathbb{R}^N$, for all $\delta > 0$, there exists a finite set of points $\{x_1, \ldots, x_r\} \subseteq I$ such that*

$$\textit{for all } x \in I, \textit{ there exists } x_i, \textit{ such that } V(x, x_i) \leq \delta. \tag{5}$$

Proof. V is continuous on $I \times I$ which is compact, therefore V is uniformly continuous on $I \times I$. Hence, for all δ, there exists ν such that

$$\forall x, x', z, z' \in I,\ \|x - x'\| \leq \nu \text{ and } \|z - z'\| \leq \nu \implies |V(x, z) - V(x', z')| \leq \delta.$$

Particularly, by setting $x' = z = z'$ and remarking that $V(x', x') = 0$, we have

$$\forall x, x' \in I,\ \|x - x'\| \leq \nu \implies V(x, x') \leq \delta.$$

Now, let us assume that for all finite set of points $\{x_1, \ldots, x_r\} \subseteq I$, there exists $x_{r+1} \in I$, such that for all x_i, $\|x - x_i\| \geq \nu$. Then, starting from a point $x_1 \in I$, we can construct a sequence $\{x_i\}_{i \in \mathbb{N}}$ such that for all $i, j \in \mathbb{N}$, $i \neq j$, we have $\|x_i - x_j\| \geq \nu$. Therefore, we cannot extract a converging subsequence of $\{x_i\}_{i \in \mathbb{N}}$ and I cannot be compact. Hence, we have proved by contradiction that there exists a finite set of points $\{x_1, \ldots, x_r\} \subseteq I$ such that for all $x \in I$, there exists x_i, such that $\|x - x_i\| \leq \nu$ which allows us to conclude (5). □

Let *Disc* be the discretization operator which maps the compact set $I \subseteq \mathbb{R}^N$ and a strictly positive number δ to a list of points $Disc(I, \delta) = \{x_1, \ldots, x_r\}$ satisfying equation (5).

Theorem 21. *Let $(x_1, y_1), \ldots, (x_r, y_r)$ be trajectories of Σ such that $Disc(I, \delta) = \{x_1(0), \ldots, x_r(0)\}$. Let $\mathcal{T}_1, \ldots, \mathcal{T}_r \in \mathcal{L}_\tau(\Sigma)$ be the associated traces. Then,*

$$(\forall i = 1, \ldots, r.\ [\![\phi]\!](\mathcal{T}_i) > \delta) \implies (\forall \mathcal{T} \in \mathcal{L}_\tau(\Sigma).\ \langle\!\langle \phi \rangle\!\rangle(\mathcal{T}) = \top)$$

Proof. Let $\mathcal{T} \in \mathcal{L}_\tau(\Sigma)$, let (x, y) be the associated trajectory of Σ. From Proposition 20, there exists $x_i(0)$ such that $V(x(0), x_i(0)) \leq \delta$. Then, from Theorem 19 and Proposition 2 from [16], it follows that $\langle\!\langle \phi \rangle\!\rangle(\mathcal{T}) = \langle\!\langle \phi \rangle\!\rangle(\mathcal{T}_i) = \top$. □

Thus, it possible to verify that the MTL property ϕ holds for all the traces of the dynamical system Σ by evaluating the robustness degree of only a finite number of simulated trajectories.

Remark 22. Similar to Theorem 21, we can prove the following statement: if for all $i = 1, \ldots, r$ it is $-[\![\phi]\!](\mathcal{T}_i) > \delta$, then for all $\mathcal{T} \in \mathcal{L}_\tau(\Sigma)$ it is $\langle\!\langle \phi \rangle\!\rangle(\mathcal{T}) = \bot$. Therefore in this case, we can conclude that all the trajectories of Σ starting in I do not satisfy the MTL specification.

4.3 Verification Algorithm

Algorithm 1 verifies that the property ϕ holds for all the traces in $\mathcal{L}_\tau(\Sigma)$. The main idea is the following. We start with a rough discretization (using a parameter $\delta > 0$) of the set of initial states – typically we pick just one point. Then, we try to verify the property using the traces associated with these initial states. When the result of the verification is inconclusive (for example when $[\![\phi]\!](\mathcal{T}_i) < \delta$), the discretization of the initial states is refined locally (using a refinement parameter $r \in (0, 1)$) around the initial states for which we were unable to conclude the property. This algorithm, therefore, allows the fast verification of robust properties, whereas more computational effort is required for non-robust properties. The refinement operation is repeated at most K times (a user defined parameter). The algorithm can terminate in one of three possible states: (i) the property has been verified for all the initial states of the system Σ, (ii) the property has been falsified (we have found a trace that does not satisfy the specification) or (iii) we have computed a subset I' of the initial states I such that all the traces initiating from I' satisfy the MTL property. In the last case, we also get a degree of coverage of the initial states that have been verified. The proof of the correctness of the algorithm is not stated here but is very similar to that of Theorem 21.

Remark 23. Let us define $\varepsilon^* := \inf_{\mathcal{T} \in \mathcal{L}_\tau(\Sigma)} |\mathbf{Dist}_\rho(\sigma, P^\phi_{\mathcal{T}})|$ to be the robustness degree of the system Σ with respect to the specification ϕ. Furthermore, consider replacing $[\![\phi]\!](\mathcal{T})$ in Algorithm 1 by the theoretical quantity $\varepsilon = \mathbf{Dist}_\rho(\sigma, P^\phi_{\mathcal{T}})$. In this case, it can be shown that whenever $\varepsilon^* > 0$, the algorithm is complete

Algorithm 1. Temporal Logic Verification Using Simulation

Input: A dynamical system $\Sigma = (N, P, f, g, I, AP, \mathcal{O})$, an MTL formula ϕ, a sequence of time stamps τ and numbers $\delta > 0$, $r \in (0, 1)$ and $K \in \mathbb{N}$.

1: **procedure** VERIFY($\Sigma, \phi, \tau, \delta, r, K$)
2: $P \leftarrow Disc(I, \delta)$, $C \leftarrow \emptyset$, $k \leftarrow 0$
3: **while** $k \leq K$ and $P \neq \emptyset$ **do**
4: $P' \leftarrow \emptyset$
5: **for** $x \in P$ **do**
6: Pick $\mathcal{T} \in \mathcal{L}_\tau(\Sigma)$ with $\sigma_0 = x$ ▷ Simulate Σ for initial state x
7: **if** $[\![\phi]\!](\mathcal{T}) < 0$ **then return** "$\mathcal{L}_\tau(\Sigma) \not\subseteq \mathcal{L}(\phi)$" ▷ ϕ does not hold on Σ
8: **else if** $[\![\phi]\!](\mathcal{T}) > r^k\delta$ **then** $C \leftarrow C \cup N_V(x, r^k\delta)$
9: **else** $P' \leftarrow P' \cup Disc(I \cap N_V(x, r^k\delta), r^{k+1}\delta)$
10: **end if** ▷ In lines 8,9: $N_V(x, \delta) = \{x' \in \mathbb{R}^N | V(x, x') \leq \delta\}$
11: **end for**
12: $k \leftarrow k + 1$, $P \leftarrow P'$
13: **end while**
14: **if** $P = \emptyset$ **then return** "$\mathcal{L}_\tau(\Sigma) \subseteq \mathcal{L}(\phi)$ " ▷ ϕ holds on Σ
15: **else return** "$\mathcal{L}_\tau(\Sigma') \subseteq \mathcal{L}(\phi)$" ▷ ϕ holds on $\Sigma' = (N, P, f, g, I \cap C, AP, \mathcal{O})$
16: **end if**
17: **end procedure**

and can verify the system using only a finite number of simulations. The current algorithm may fail to be complete since we are using an under-approximation of the robustness degree (note also that $[\![\phi]\!](\mathcal{T}) = 0 \not\Rightarrow \mathbf{Dist}_\rho(\sigma, P^\phi_\mathcal{T}) = 0$).

5 Experimental Results

In this section, we demonstrate the applicability of our framework through some experimental results. We have implemented Algorithm 1 in MATLAB for linear dynamical systems and applied it to the problem presented in Example 7. A bisimulation function of the form $V(x_1, x_2) = \sqrt{(x_1 - x_2)^T M (x_1 - x_2)}$, where M is a positive definite symmetric matrix, has been computed following the technique described in [25]. We run the verification algorithm for $T \in \{0.8, 1.2, 1.6\}$ and $\theta \in \{1.4, 1.5, 1.6\}$. The results are summarized in Table 1.

Table 1. Experimental results of the verification algorithm for the transmission line example. For each value of (T, θ) the table gives whether the property ϕ holds on Σ and how many simulations of the system were necessary to conclude.

	$T = 0.8$	$T = 1.2$	$T = 1.6$
$\theta = 1.4$	False / 1	False / 7	False / 7
$\theta = 1.5$	False / 1	True / 15	True / 9
$\theta = 1.6$	False / 1	True / 15	True / 7

We can see that even though our algorithm does not have a completeness result, we were able in all the cases we considered to verify or falsify the property. The number of simulations needed for the verification depends on the value of the parameters T and θ. For instance, for the value $T = 0.8$, we were able to falsify the property using only one simulation, independently of the value of θ. For $\theta = 1.4$, the property is also false independently of the value of T. However, for $T = 1.2$ and $T = 1.6$, we needed 7 simulations of the system to falsify the property. Essentially, this means that the properties $\Box\pi_1$ with $\mathcal{O}(\pi_1) = [-1.4, 1.4]$ and $\Diamond_{[0,0.8]}\Box\pi_2$ are both false on Σ but the former much less robustly than the latter. For the cases where the property ϕ holds on Σ, we can see that the number of simulations needed for the verification is also related to the robustness of the system with respect to the property. Indeed, the larger the T and θ are, the more robust the Σ is with respect to the property ϕ and, in turn, the less is the number of the simulations that are required for verification. This is one interesting feature of our approach which relates robustness to the computational complexity of the verification.

6 Conclusions and Future Work

We have presented a novel approach to the verification problem of temporal properties of continuous time dynamical systems. Our framework reinforces a very intuitive observation: robustly (safe or unsafe) systems are easier to verify. We believe that light weight verification methods, such as the one presented here, can offer valuable assistance to the practitioner. This line of work can be extended to multiple fronts. One important direction, as advocated in [18,19,20], is to relax the requirement that all the traces should have the same sequence of time stamps. Another direction is to move toward the simulation based verification of hybrid and stochastic systems.

Acknowledgments. The authors are grateful to Zhi Han for providing them with the transmission line example. This work has been partially supported by NSF ITR Grant 0121431, NSF EHS Grant 0311123, NSF PECASE Grant 0132716 and the European community project IST-2003-507219 PROSYD (Property-based System Design).

References

1. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. MIT Press, Cambridge, Massachusetts (1999)
2. Alur, R.: Timed automata. In: Proceedings of the 11th Computer Aided Verification. Volume 1633 of LNCS., Springer (1999) 8–22
3. Asarin, E., Bournez, O., Dang, T., Maler, O.: Approximate reachability analysis of piecewise linear dynamical systems. In: Hybrid Systems: Computation and Control. Volume 1790 of LNCS., Springer (2000) 21–31

4. Asarin, E., Dang, T., Girard, A.: Reachability of non-linear systems using conservative approximations. In: Hybrid Systems: Computation and Control. Volume 2623 of LNCS., Springer (2003) 22–35
5. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theoretical Computer Science **138** (1995) 3–34
6. Chutinan, A., Krogh, B.: Verification of polyhedral invariant hybrid automata using polygonal flow pipe approximations. In: Hybrid Systems: Computation and Control. Volume 1569 of LNCS., Springer (1999) 76–90
7. Henzinger, T.A., Ho, P.H., Wong-Toi, H.: Algorithmic analysis of nonlinear hybrid systems. IEEE Transactions on Automatic Control **43** (1998) 540–554
8. Frehse, G.: Phaver: Algorithmic verification of hybrid systems past hytech. In: Hybrid Systems: Computation and Control. Volume 3414 of LNCS., Springer (2005) 258–273
9. Girard, A.: Reachability of uncertain linear systems using zonotopes. In: Hybrid Systems: Computation and Control. Volume 3414 of LNCS. (2005) 291–305
10. Han, Z.: Formal Verification of Hybrid Systems using Model Order Reduction and Decomposition. PhD thesis, Dept. of ECE, Carnegie Mellon University (2005)
11. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: Proceedings of FORMATS-FTRTFT. Volume 3253 of LNCS. (2004) 152–166
12. Kapinski, J., Krogh, B.H., Maler, O., Stursberg, O.: On systematic simulation of open continuous systems. In: Hybrid Systems: Computation and Control. Volume 2623 of LNCS., Springer (2003) 283–297
13. Esposito, J.M., Kim, J., Kumar, V.: Adaptive RRTs for validating hybrid robotic control systems. In: International Workshop on the Algorithmic Foundations of Robotics. (2004)
14. Girard, A., Pappas, G.J.: Verification using simulation. In: Hybrid Systems: Computation and Control (HSCC). Volume 3927 of LNCS., Springer (2006) 272 – 286
15. Koymans, R.: Specifying real-time properties with metric temporal logic. Real-Time Systems **2** (1990) 255–299
16. Fainekos, G.E., Pappas, G.J.: Robustness of temporal logic specifications for finite state sequences in metric spaces. Technical Report MS-CIS-06-05, Dept. of CIS, Univ. of Pennsylvania (2006)
17. Girard, A., Pappas, G.J.: Approximation metrics for discrete and continuous systems. Technical Report MS-CIS-05-10, Dept. of CIS, Univ. of Pennsylvania (2005)
18. Huang, J., Voeten, J., Geilen, M.: Real-time property preservation in approximations of timed systems. In: Proceedings of the 1st ACM & IEEE International Conference on Formal Methods and Models for Co-Design. (2003) 163–171
19. Henzinger, T.A., Majumdar, R., Prabhu, V.S.: Quantifying similarities between timed systems. In: FORMATS. Volume 3829 of LNCS., Springer (2005) 226–241
20. Alur, R., Torre, S.L., Madhusudan, P.: Perturbed timed automata. In: Hybrid Systems: Computation and Control. Volume 3414 of LNCS. (2005) 70–85
21. de Alfaro, L., Faella, M., Henzinger, T.A., Majumdar, R., Stoelinga, M.: Model checking discounted temporal properties. In: Tools and Algorithms for the Construction and Analysis of Systems. Volume 2988 of LNCS., Springer (2004) 77–92
22. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes: The Art of Scientific Computing. 2nd edn. Cambridge University Press, Cambridge (UK) and New York (1992)
23. Ouaknine, J., Worrell, J.: On the decidability of metric temporal logic. In: 20th IEEE Symposium on Logic in Computer Science (LICS). (2005) 188–197

24. Thati, P., Rosu, G.: Monitoring algorithms for metric temporal logic specifications. In: Runtime Verification. Volume 113 of ENTCS., Elsevier (2005) 145–162
25. Girard, A., Pappas, G.J.: Approximate bisimulations for constrained linear systems. In: Proceedings of 44th IEEE Conference on Decision and Control and European Control Conference. (2005) 4700–4705
26. Girard, A., Pappas, G.J.: Approximate bisimulations for nonlinear dynamical systems. In: Proceedings of 44th IEEE Conference on Decision and Control and European Control Conference. (2005) 684–689