Computing Descent Direction of MTL Robustness for Non-Linear Systems

Houssam Abbas and Georgios Fainekos

Abstract— **The automatic analysis of transient properties of nonlinear dynamical systems is a challenging problem. The problem is even more challenging when complex state-space and timing requirements must be satisfied by the system. Such complex requirements can be captured by Metric Temporal Logic (MTL) specifications. The problem of finding system behaviors that do not satisfy an MTL specification is referred to as MTL falsification. This paper presents an approach for improving stochastic MTL falsification methods by performing local search in the set of initial conditions. In particular, MTL robustness quantifies how correct or wrong is a system trajectory with respect to an MTL specification. Positive values indicate satisfaction of the property while negative values indicate falsification. A stochastic falsification method attempts to minimize the system's robustness with respect to the MTL property. Given some arbitrary initial state, this paper presents a method to compute a descent direction in the set of initial conditions, such that the new system trajectory gets closer to the unsafe set of behaviors. This technique can be iterated in order to converge to a local minimum of the robustness landscape. The paper demonstrates the applicability of the method on some challenging nonlinear systems from the literature.**

I. Introduction

A number of applications can only be accurately modeled using nonlinear dynamical models. Typical such applications include analog circuits [1]–[3] and biological and medical systems [4]–[7]. A common theme of all the aforementioned applications is the need to verify transient or periodic properties of the system. Such properties might involve sequencing of events, conditional reachability and invariants and real-time constraints and can be formally captured using temporal logics [4], [8].

Unfortunately, for complex nonlinear systems, these types of properties are hard – if not impossible – to verify algorithmically. Therefore, recent research efforts have been invested in property falsification methods [9]–[12]. In falsification, the space of operating conditions and/or inputs is searched in order to find an initial condition and/or parameter that will force the system to exhibit an unsafe behavior with respect to the formal requirement. In turn, the unsafe system trajectory can be used in order to manually or automatically modify the system to achieve the desired system behavior and performance [13], [14].

In [10], [15], the temporal logic falsification problem is converted into an optimization (minimization) problem based on the notion of robustness of temporal logics [16]. Essentially, a system trajectory with negative robustness is one that proves the existence of unsafe system behaviors. Then, a number of stochastic optimization methods can be utilized in order to solve the optimization problem and find a system trajectory that minimizes the temporal logic robustness metric.

However, in [10], [15], the system is treated as a black-box. In order, to improve the rate of convergence of stochastic search methods, it is desirable to have techniques that can compute local descent directions in the search space. In particular, if a test is performed starting from an initial condition x with property robustness $f(x)$, then a descent vector d must be computed so that starting from $x+d$ the system has robustness $f(x+d) < f(x)$. Such a process has the potential to speed up the stochastic search method by enabling gradient descent in the search space. In [17], we demonstrated that in the case of linear hybrid systems improvements in the convergence rate can be achieved.

Contributions: In this paper, we present a method for the computation of descent vectors for reducing specification robustness for continuous nonlinear dynamical systems. In particular, given an arbitrary Metric Temporal Logic (MTL) specification [18], we determine a critical point on the system trajectory which if changed, then the MTL robustness will be changed as well. We utilize nonsmooth optimization theory [19] in order to derive the equations that compute a descent vector in the set of initial conditions that will result in reduced MTL robustness. Finally, we demonstrate the applicability of our approach on some nonlinear models from the literature. We envision that our results can be extended to handle arbitrary temporal logic specifications over trajectories of hybrid systems.

Related Work: Combined state-space and real-time temporal logic properties have been studied in a number of different settings. MTL properties of nonlinear systems have been studied in [12] through abstractions to Linear Parameter Varying (LPV) systems. The work in [11] studies the applicability of statistical model checking methods on stochastic hybrid systems. The temporal logic falsification problem can be viewed as a dual problem to the optimal control problem under temporal logic requirements. In [20], the optimal control problem under Linear Temporal Logic (LTL) specifications is studied for mixed-logical discrete-time linear dynamical systems. However, there do not exist any optimal control problem formulations for nonlinear systems under MTL specifications.

The work that appears in [4] and [21] is the closest to the results that we present here. In particular, in [21], the authors use sensitivity analysis in order to quantify neighborhoods of trajectories with the same qualitative behavior. Then, the results of [21] are extended in [4] to estimating parameter

This work was partially supported by the NSF awards CNS-1017074 and CNS-1116136.

H. Abbas and G. Fainekos are with the Schools of Engineering at Arizona State University, Tempe, AZ, E-mail: {hyabbas, fainekos}@asu.edu

ranges and initial conditions for which the system satisfies some real-time temporal logic specification. Even though we are also using sensitivity analysis in our problem solution, our objective is very different from the work in [4]. Our goal is to develop the local search tools needed in order to improve the performance of stochastic MTL falsification methods [10], [15]. Stochastic falsification methods avoid the state-explosion problems that occur when attempting to cover a high-dimensional set of parameters.

II. Problem Formulation

We consider a dynamical system with state $x \in X$

$$\dot{x} = F(t, x) \tag{1}$$

for a C^1 flow $F : \mathbb{R}^n \to \mathbb{R}^n$ with initial conditions $x_0 \in X_0$.

Assumption 2.1: For every $x \in X_0$ and finite time $T > 0$, there exists a unique solution $s(\cdot, x) : [0, T] \mapsto \mathbb{R}^n$ to the differential equation (1). Also, the solution $s_x(\cdot)$ is absolutely continuous. Finally, the flow F is locally bounded, that is, for all compact sets $[0, t] \times C \subset [0, T] \times X_0$, there exists $m > 0$ such that $F([0, T] \times C) \subset m\mathbf{B}$, where $\mathbf{B}$ is the unit ball centered at 0.

We formally capture specifications regarding the correct system behavior using Metric Temporal Logic (MTL) [18]. MTL formulas are built over a set of propositions using combinations of the traditional and temporal operators. In this work, the set of atomic propositions AP label subsets of the state space X. In other words, we define an observation map $\mathcal{O} : AP \to \mathcal{P}(X)$ such that for each $\pi \in AP$ the corresponding set is $\mathcal{O}(\pi) \subseteq X$. Here, $\mathcal{P}(S)$ denotes the powerset of a set S. Traditional logic operators are the *conjunction* ($\wedge$), *disjunction* ($\vee$), *negation* ($\neg$), *implication* ($\to$) and *equivalence* ($\leftrightarrow$). Some of the temporal operators are *eventually* ($\Diamond_{\mathcal{I}}$), *always* ($\Box_{\mathcal{I}}$) and *until* ($U_{\mathcal{I}}$). The subscript $\mathcal{I}$ imposes timing constraints on the temporal operators. The interval $\mathcal{I}$ must be non-empty ($\mathcal{I} \neq \emptyset$). For example, MTL can capture the requirement that "all the trajectories $x(t) \in \mathbb{R}$ attain a value in the set $[10, +\infty)$" ($\Diamond p_1$ with $\mathcal{O}(p_1) = [10, +\infty)$) or that "whenever the value of x drops below 10, then it should go above 10 within $5\,\mathrm{sec}$ and remain above 10 for at least $10\,\mathrm{sec}$" ($\Box(\neg p_1 \to \Diamond_{[0,5]}\Box_{[0,10]} p_1)$).

We can quantify how robustly a system trajectory $s_x(t) = s(t, x)$ satisfies a specification ϕ in MTL [16]. Namely, we define a function $f_\phi(x)$ that returns the radius of the largest neighborhood we can fit around s_x such that any trajectory in that neighborhood satisfies the same MTL specification ϕ as s_x. Moreover, $f_\phi(x)$ takes positive values if s_x satisfies ϕ and negative values otherwise. The falsification of specification ϕ, i.e. detecting a system behavior that does not satisfy ϕ, can thus be re-cast as the problem of finding initial states $x \in X_0$ with negative f_ϕ-values. This can be done using stochastic search techniques [10], [15]. These can be improved by computing local descent directions for f_ϕ.

In this paper, our objective is to solve the following subproblem: Let $\mathcal{U} \subset X$ be a set of 'unsafe' system states - in the next section we see exactly what such a $\mathcal{U}$ looks like. There may be many such sets. We define the robustness of a trajectory relative to $\mathcal{U}$:

Definition 2.1 (Robustness): Let $x \in X_0$, $T > 0$ and $s_x(\cdot)$ be the unique solution of (1) starting from time 0, then the robustness of the solution s_x with respect to $\mathcal{U}$ is

$$f(x) = \min_{0 \leq t \leq T} d_{\mathcal{U}}(s(t; x)) \tag{2}$$

where $d_{\mathcal{U}}(x) = \inf_{u \in \mathcal{U}} \|x - u\|$ is the distance function of a point x from $\mathcal{U}$.

The function f is non-differentiable, and generally non-convex. Then, our problem is:

Problem 1: Given $x \in X_0$, $T > 0$ and the unsafe set $\mathcal{U}$, find a vector $d(x) \in \mathbb{R}^n$ such that

$$f(x + hd(x)) < f(x) \text{ for all } 0 < h < \overline{h}$$

for some $\overline{h} > 0$.

Although Problem 1 was defined for a single unsafe set, Prop. 3.1 below shows that robustness w.r.t. a general MTL formula (with several sets) equals the robustness w.r.t. one of the formula's atomic propositions (one of the sets).

Some proofs are omitted due to space constraints.

III. MTL Robustness

In this section, we provide an informal review of the robust semantics of MTL formulas. Formal details are available in our previous work [16].

Definition 3.1 (MTL Syntax): Let AP be the set of atomic propositions and $\mathcal{I}$ be any non-empty interval of $\mathbb{R}_{\geq 0}$. The set MTL of all well-formed MTL formulas is inductively defined as $\varphi ::= \mathbf{T} \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \mathcal{U}_{\mathcal{I}} \varphi$, where $p \in AP$ and $\mathbf{T}$ is *true*.

The robust semantics maps an MTL formula φ and a trajectory s_x to a value drawn from $\mathbb{R} \cup \{\pm\infty\}$. The semantics for the atomic propositions evaluated for $s_x(t)$ consists of the distance between $s_x(t)$ and the set $\mathcal{O}(p)$ labeling atomic proposition p. Intuitively, this distance represents how robustly the point $s_x(t)$ lies within (or is outside) the set $\mathcal{O}(p)$. If this distance is zero, then the smallest perturbation of the point $s_x(t)$ can affect the outcome of $s_x(t) \in \mathcal{O}(p)$. The semantics for a formula are naturally defined from the semantics for the atomic propositions. We denote the robust valuation of the formula φ over the trajectory s_x at time t starting at initial condition x by $[\![\varphi, \mathcal{O}]\!](s_x, t)$. It is easy to show [16] that if the signal satisfies the property, then its robustness is non-negative, and if the signal does not satisfy the property, then its robustness is non-positive. In [8], we presented algorithms for efficiently computing the MTL robustness of a discrete-time trajectory. The analysis can be extended to continuous-time signals under some assumptions on the system [16].

For computational reasons, we must impose additional assumptions on the sets $\mathcal{O}(p)$:

Assumption 3.1: For each $p \in AP$, we have $\mathcal{O}(p) = \cap_i \{x \in \mathbb{R}^n \mid a_i \cdot x \leq b_i\}$ where $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$.

Under the assumption that (1) is well-behaved, there exist at least one point in time t and an atomic proposition p such that the MTL robustness is equal to the distance of $s_x(t)$ from $\mathcal{O}(p)$. The proof of the following proposition is based on the assumption that the trajectory is continuous and bounded for all time in $[0, T]$.

Proposition 3.1: Consider an MTL formula ϕ and a trajectory s_x of (1) starting from some $x \in X_0$ such that $[\![\phi, \mathcal{O}]\!](s_x, 0) > 0$. If (1) satisfies Assumption 2.1, then there exist $t_r \in [0, T]$ and $p \in AP$ such that

$$[\![\phi, \mathcal{O}]\!](s_x, 0) = \mathbf{Dist}(s_x(t_r), \mathcal{O}(p))$$

where the *signed* distance $\mathbf{Dist}(z, S) = d_S(z)$ if $z \in S$, and $-d_S(z)$ otherwise. We remark that given a trajectory of (1), then the sample of the trajectory that represents the critical distance can be easily computed by modifying the algorithm in [8].

In order to detect a bad system behavior with respect to an MTL specification, our goal is to reduce such critical distances. Therefore, in the following, we focus on a particular set $\mathcal{O}(p)$ or one of its defining half-spaces which we refer to as the unsafe set $\mathcal{U}$.

In general, ϕ may have several predicates p and corresponding sets $\mathcal{O}(p)$. To falsify ϕ will require finding a trajectory that visits these $\mathcal{O}(p)$ in some order and under some timing constraints. In this paper, we derive the descent vector relative to only one $\mathcal{O}(p)$ at a time. Different unsafe sets $\mathcal{O}(p)$ are chosen by the stochastic falsification algorithm, which calls the local descent algorithm on the chosen unsafe set.

IV. Computing a Descent direction

In this section we compute a descent direction using tools from nonsmooth analysis. We start by solving the unconstrained problem $X_0 = \mathbb{R}^n$ in sub-section IV-A. The constrained problem is later addressed in sub-section IV-B.

A. Descent vector

In general, two trajectories starting arbitrarily close may achieve very different robustness values, at very different points in time. The following theorem shows that for some systems that are themselves 'Lipschitz' (in the sense below), the objective function is Lipshitz:

Theorem 4.1 (Lipschitz objective): If for every $x \in X_0$, there exist $b > 0$ and $K_x > 0$ s.t. $\|s(t; x_1) - s(t; x_2)\| \leq K_x\|x_1 - x_2\|$ for all $x_1, x_2 \in B_b(x)$ and all $0 \leq t \leq T$, then the objective function f is Lipschitz.
The condition of the theorem can be shown to hold if we assume F to be Lipschitz in x on $[0, T] \times X$, and X is open connected. Moreover, the constant K_x is then independent of x.

Nonsmooth analysis [19] provides us with the tools to compute descent directions.

Theorem 4.2 (Thm. 5.2.5 in [19]): Let $f : \mathbb{R}^n \to$ be locally Lipschitz at x. The direction $d \in \mathbb{R}^n$ is a descent direction at x if

$$f^o(x; d) < 0$$

where f^o is the generalized directional derivative of f at x

$$f^o(x; d) = \limsup_{y \to x, h \searrow 0} \frac{f(y + hd) - f(y)}{h}$$

Theorem 4.3 (2.1.3(i) in [19]): Let $g : \mathbb{R}^n \to$ be a convex function with a Lipschitz constant K at x. Then, the directional derivative in each direction $v \in \mathbb{R}^n$ exists and satisfies

$$g'(x; v) = \inf_{h>0} \frac{g(x + hv) - g(x)}{h}$$

In this section we will work from the definition of generalized derivative to derive a descent d such that $f^o(x; d) < 0$.

By definition of robustness (2), we have

$$\begin{aligned} f^o(x; d) &= \limsup_{y \to x, h \searrow 0} \frac{f(y + hd) - f(y)}{h} \\ &= \limsup_{y \to x, h \searrow 0} \Big(\min_{0 \leq t \leq T} d_{\mathcal{U}}(s(t; y + hd)) - \\ &\quad - \min_{0 \leq t \leq T} d_{\mathcal{U}}(s(t; y)) \Big) / h \end{aligned}$$

By definition of limit, there exists sequences $(y_i) \to x \in \mathbb{R}^n$ and $(h_i) \to 0 \in \mathbb{R}_+$ and $i_0 \in \mathbb{N}$ such that, for $i > i_0$,

$$\begin{aligned} f^o(x; d) \leq \Big(&\min_{0 \leq t \leq T} d_{\mathcal{U}}(s(t; y_i + h_i d)) \\ &- \min_{0 \leq t \leq T} d_{\mathcal{U}}(s(t; y_i)) \Big) / h_i + \frac{1}{i} \end{aligned}$$

It is easily seen that for positive functions $g(t)$ and $k(t)$, $\min_t g(t) - \min_t k(t) \leq -\min_t[k(t) - g(t)]$. Identifying $g(t) = d_{\mathcal{U}}(t; y_i + h_i d)$ and $k(t) = d_{\mathcal{U}}(t; y_i)$, we have

$$\begin{aligned} &f^o(x; d) \leq \\ &\leq \frac{-\min_{0 \leq t \leq T}[d_{\mathcal{U}}(s(t; y_i)) - d_{\mathcal{U}}(s(t; y_i + h_i d))]}{h_i} + \frac{1}{i} \\ &= -\min_{0 \leq t \leq T} \frac{[d_{\mathcal{U}}(s(t; y_i)) - d_{\mathcal{U}}(s(t; y_i + h_i d))]}{h_i} + \frac{1}{i} \end{aligned}$$

As $i \to \infty$, $1/i \to 0$, $h_i \to 0$, $y_i \to x$ and $s(t; y_i + h_i d) \to s(t; y_i)$ in norm by Assumption 2.1. So

$$\begin{aligned} &f^o(x; d) \\ &\leq \lim_{i \to \infty} \left\{ -\min_{0 \leq t \leq T} \frac{[d_{\mathcal{U}}(s(t; y_i)) - d_{\mathcal{U}}(s(t; y_i + h_i d))]}{h_i} \right\} \\ &= -\min_{0 \leq t \leq T} \lim_{i \to \infty} \frac{[d_{\mathcal{U}}(s(t; y_i)) - d_{\mathcal{U}}(s(t; y_i + h_i d))]}{h_i} + \frac{1}{i} \\ &= -\min_{0 \leq t \leq T} \lim_{y_i \to x, h_i \searrow 0} -\frac{d_{\mathcal{U}}(s(t; y_i + h_i d)) - d_{\mathcal{U}}(s(t; y_i))]}{h_i} \end{aligned}$$

(We can show that the interchange of limit and min above is valid). Linearizing $s(t; y_i + h_i d)$ in the second argument, and ignoring higher-order terms $o(h_i)$:

$$s(t; y_i + h_i d) \approx s(t; y_i) + h_i \frac{\partial s(t; y_i)}{\partial y} d \qquad (3)$$

Assumption 4.1: We assume that the sensitivity matrix $A(t; y) \triangleq \frac{\partial s(t;y)}{\partial y}$ exists, is invertible, and that it is spectral norm-continuous in y.

We remark that $A(t;y)$ is the sensitivity of the trajectory with respect to the initial conditions and can be computed as indicated in [22], [23]. Then,

$$\begin{aligned} &f^o(x;d) \leq \\ &\leq -\min_{0\leq t\leq T}[-\lim_{y_i\to x, h_i\searrow 0}(d_{\mathcal{U}}(s(t;y_i)+h_iA(t;y_i)d)- \\ &- d_{\mathcal{U}}(s(t;y_i)))/h_i] \end{aligned}$$

If the limit in brackets does not exist, i.e., it is $+\infty$, then $f^o(x;d) < 0$ and we are done. Otherwise, it can be shown that the limit in brackets equals $d'_{\mathcal{U}}(s(t;A(t;x)d)$: that is, the directional derivative of $d_{\mathcal{U}}$ at $s(t;x) \in \mathbb{R}^n$, in the direction $A(t;x)d$. Thus,

$$\begin{aligned} f^o(x;d) &\leq -\min[-d'_{\mathcal{U}}(s(t;x);A(t;x)d)] \\ &= \max_{0\leq t\leq T} d'_{\mathcal{U}}(s(t;x);A(t;x)d) \end{aligned}$$

Recall that we want $f^o(x;d) < 0$, so we seek to upper-bound the RHS, that is,

$$\max_{0\leq t\leq T} d'_{\mathcal{U}}(s(t;x);A(t;x)d) < 0,$$

which is equivalent to

$$d'_{\mathcal{U}}(s(t;x);A(t;x)d) < 0 \;\forall\; t \in [0,T]$$

Fix t for now. For ease of notation, we'll just write s and A for $s(t;x)$ and $A(t;x)$, respectively. By Theorem 4.3,

$$d'_{\mathcal{U}}(s;Ad) = \inf_{h>0} \frac{d_{\mathcal{U}}(s+h\cdot Ad) - d_{\mathcal{U}}(s)}{h}$$

Thus, it is necessary that there exist an $h > 0$ s.t.

$$d_{\mathcal{U}}(s+h\cdot Ad) - d_{\mathcal{U}}(s) < 0$$

Let $n_{s(x)}(t) \in \mathbb{R}^n$ be the vector that gives the direction of the shortest distance between $s(t;x)$ and $\mathcal{U}$. We'll write n for short, and call it an *approach vector*. Then

$$d_{\mathcal{U}}(s+hn) < d_{\mathcal{U}}(s) \;\forall\; 0 < h \leq d_{\mathcal{U}}(s) \Rightarrow \qquad (4)$$

$$\inf_{h>0} \frac{d_{\mathcal{U}}(s+hn) - d_{\mathcal{U}}(s)}{h} < \frac{d_{\mathcal{U}}(s+hn) - d_{\mathcal{U}}(s)}{h} < 0$$

So set $A(t;x)d(t) = n_{s(x)}(t) \Rightarrow d(t) = A(t;x)^{-1}n_{s(x)}(t)$, where we made explicit the dependence of the descent vector on time (different points on the trajectory will have different descent vectors). Thus, $d(t) = A(t;x)^{-1}n_{s(x)}(t)$ satisfies $d'_{\mathcal{U}}(s(t;x);A(t;x)d(t)) < 0$ at every t. In particular at

$$t^* \triangleq \text{argmax}_{0\leq t\leq T} d'_{\mathcal{U}}(s(t;x);A(t;x)d(t)),$$

we still have

$$d'_{\mathcal{U}}(s(t^*;x);A(t^*;x)d(t^*) < 0$$

Finally,

$$d = A(t^*;x)^{-1}n_{s(x)}(t^*) \qquad (5)$$

is a descent direction for f at x, subject to the foregoing assumptions.

It remains to compute t^*. We can show that $t^* = \text{argmin}_{0\leq t\leq T} d_{\mathcal{U}}(s(t;x))$, and the proof is omitted.

We conclude this section by noting that Eq.(5) can be generalized by choosing a different approach vector than n, conditioned on satisfying (4). The particular choice will depend on the geometry of the problem.

B. Constrained problem

We now remove Assumption (A3) and we consider the constrained problem where $X_0 \neq \mathbb{R}^n$. In other words, what if $x + d \notin X_0$?

If we use $\mu d, \mu < 1$, then

$$\begin{aligned} &d'_{\mathcal{U}}(s(t_r;x);\mu\cdot n_{s(x)}(t_r)) = \\ &= \inf_{h>0} \frac{d_{\mathcal{U}}(s(t_r;x)+h\mu\cdot d_{s(x)}(t)) - d_{\mathcal{U}}(s(t_r;x))}{h} < 0 \end{aligned}$$

by Eq. (4). So we can shrink d to fit $x + d$ in X_0, and still have a descent. This simple approach circumvents the need to calculate or approximate the subdifferential of f subject to the constraints, which is a non-trivial task given the form of f.

This brings up the issue of step-size: in principle, any method for computing a step-size, that does not require differentiability, can be used, once we have a descent direction (and indeed we use backtracking in our experiments); see e.g. [19], [24], [25]. In practice, a method that does not use a line-search is preferable, since line searches require additional evaluations of the objective function, and this implies simulating the system. Such simulations might prove too costly. We will simply highlight two requirements on any step-size that transpire from above arguments: that it be "small enough" for the $o(h)$ terms in (3) to be safely ignored, and that it be smaller than the robustness $d_{\mathcal{U}}(s(t;x))$ as per (4). Additional, generic, conditions can be reviewed in standard texts, such as [19, Section II.2.1.2].

V. Experiments

Example 1: Our first example is 2-dimensional system taken from [12] (Example 4), given by

$$\dot{x}(t) = \begin{bmatrix} 0.05x_1(t)\sin^2(x_2(t)) - 2.5x_2(t) \\ 0.5x_1(t) - x_2(t) \end{bmatrix}$$

We present two representative experiments with this system, both using a trajectory duration of 10 time units, the specification is $\Box\neg p_1$ with $\mathcal{O}(p_1) = [-0.11, -0.08] \times [0, 0.01]$ and $x_0 = (0.5, -0.2)^T$. First, we consider $h = 1$. Fig.1 shows a sequence of starting points, and corresponding trajectories, generated by computing successive descent vectors according to Eq. (5). Descents of different directions are generated, and successive trajectories get closer to the unsafe set as can be seen in Fig.2. Ten descent vectors reduce robustness from 0.016097 to 0.011181.

Starting with $h = 0.1$, the iterations reach a local minimum after 4 descents - the d computed by Eq.(5) no longer decreases the objective function value for any step-size. A small ball around the current x_0 was sampled to verify it is indeed a local minimum. $\triangle$

Example 2: Our second example is taken from [15], given by

$$\dot{x}(t) = \begin{bmatrix} x_1(t) - x_2(t) + 0.1t \\ x_2(t)\cos(2\pi x_2(t)) - x_1(t)\sin(2\pi x_1(t)) + 0.1t \end{bmatrix}$$

with initial condition $x(0) = x_0 \in X_0 = [-1,1] \times [-1,1]$, and specification $\Box\neg p_2$ with $\mathcal{O}(p_2) = [-1.6, -1.4] \times$

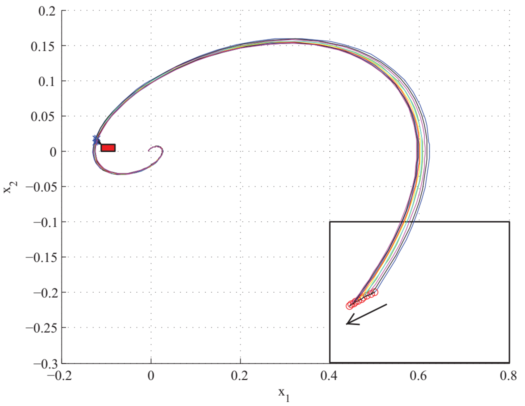

Fig. 1. Inital set (bottom right), unsafe set (red (black) box in top left) and trajectories for Example 1.

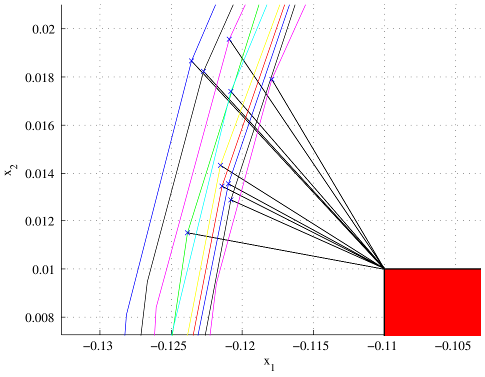

Fig. 2. Successive Example 1 trajectories descending towards unsafe set.

$[-0.9, -1.1]$. If the trajectory duration is 6 units, allowing the trajectories to settle, and starting from $(0, 0)^T$, a local minimum is reached in only 2 iterations. Inspection of the descent direction lead us to try a start point $x_0 = (0.5, 0.5)^T$: from here, robustness was reduced from 1.9 to 1.19 in 10 iterations, decreasing at every iteration. If the trajectory duration is only 2 units, thus remaining in the transient mode, we can see more clearly the effect of choosing a descent direction: Fig.3 shows the unsafe set relative to the initial set, and the trajectories chosen by descent.

To verify that this change in trajectory was not 'accidental' (e.g. as a result of the step-size leading to an entirely different local min), but rather was driven by a genuine descent, we plot the contour curves of the objective function (obtained by sampling it on a grid of 500 points). Fig.4 shows a consistent descent towards levels of decreasing robustness. As further verification, we moved the unsafe set to $[1.251.75] \times [-1.1 - 0.9]$. Fig.5 shows the resulting trajectories chosen by descent.

In order to demonstrate the potential of the proposed approach to the MTL falsification problem, we incorporated the descent method with the Simulated Annealing (SA) falsification method of [15]. We falsified the specification

$$\phi_3 = \Box(p_3 \implies \Box_{[0,1]} \neg p_4)$$

where $\mathcal{O}(p_3)$ and $\mathcal{O}(p_4)$ are the dark boxes in Fig.6. Informally, the specification requires that if the system trajectory is in $\mathcal{O}(p_3)$ at time t_1, then $\mathcal{O}(p_4)$ should be avoided for all time in $[t_1, t_1 + 1]$. For the specification to be falsified distances to both sets $\mathcal{O}(p_3)$ and $\mathcal{O}(p_4)$ must become zero. Note that in Fig. 6, our algorithm attempts to minimize both distances. To rigorously assess the efficiency of SA+DESCENT compared to pure SA, a thorough statistical study will be conducted in future research. $\triangle$

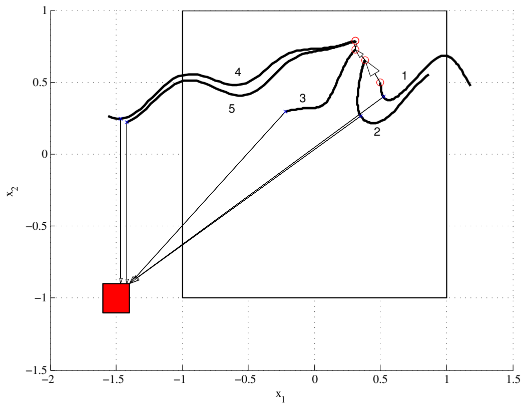

Fig. 3. Transient trajectories of Example 2. Note the qualitative change in the trajectories, from 1 to 5, as a result of descending towards the unsafe set. Circles mark the initial points, and long black arrows are $u^* - s(t; x)$.

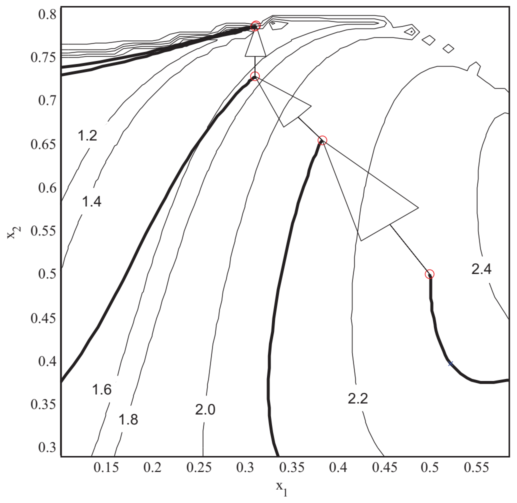

Fig. 4. Contour plot of f in Example 2, with initial points chosen by descent.

Example 3: Our third example is the quorum sensing system of the luminescent marine bacterium *Vibrio Fischeri* (VF) [5]. This is modeled as a 9-dimensional non-linear system. A simplified hybrid model of a mutant VF bacterium has 2 equilibrium points (one luminescent, the other non-luminescent) [5]. We choose the unsafe set to be disjoint from neighborhoods around these 2 equilibria. Namely, we consider the specification $\Box\neg p_2$ with $\mathcal{O}(p_5) = \{x \in \mathbb{R}^9 \mid 13625 \leq x_3 \leq 13626, 36330 \leq x_7 \leq 36331, 17968 \leq x_8 \leq 17969\}$. Starting from $x_0 = (1e5, 1, \ldots, 1)^T$, and

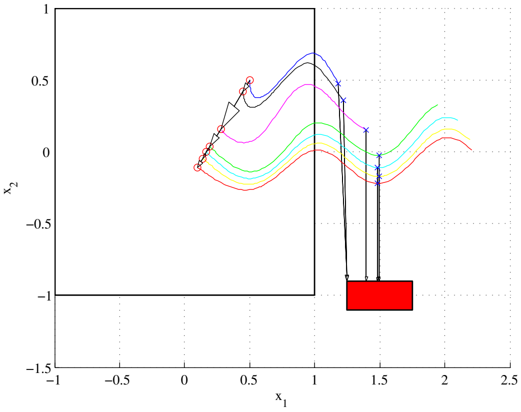

Fig. 5. Example 2 with a different unsafe set.

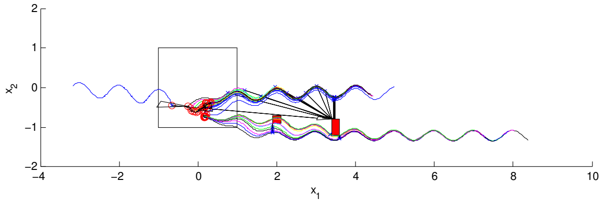

Fig. 6. Example 2 with ϕ_3. $\mathcal{O}(p_3)$ is the left dark square, $\mathcal{O}(p_4)$ is the right dark square, X_0 is the white rectangle.

computing trajectories of duration 5 units, 10 computations of a descent vector with step size $h = 0.1$ reduce robustness from 36327 to 14099, with robustness decreasing at each iteration. △

VI. Conclusions

We have presented the derivation of the equations that can be used for the computation of robustness descent vectors in the set of initial conditions for nonlinear dynamical systems. These results are necessary for enabling "gray box" MTL falsification methods for dynamical systems. In the future, we will focus on extending our new approach to hybrid systems and non-autonomous systems.

References

[1] S. Steinhorst and L. Hedrich, "Model checking of analog systems using an analog specification language," in *Proceedings of the conference on Design, automation and test in Europe*, ser. DATE '08. New York, NY, USA: ACM, 2008, pp. 324–329.

[2] M. H. Zaki, S. Tahar, and G. Bois, "Formal verification of analog and mixed-signal designs: A survey," *Microelectronics Journal*, vol. 39, p. 13951404, 2008.

[3] S. Little, D. Walter, K. Jones, and C. J. Myers, "Analog/mixed-signal circuit verification using models generated from simulation traces," in *Proceedings of the 5th International Symposium on Automated Technology for Verification and Analysis (ATVA)*, ser. LNCS, vol. 4762. Springer, 2007, pp. 114–128.

[4] A. Donze, E. Fanchon, L. M. Gattepaille, O. Maler, and P. Tracqui, "Robustness analysis and behavior discrimination in enzymatic reaction networks," *PLoS ONE*, vol. 6, no. 9, p. e24246, 09 2011.

[5] C. Belta, J. Schug, T. Dang, V. Kumar, G. Pappas, and H. Rubin, "Stability and rechability analysis of a hybrid model of luminescence in the marine bacterium vibrio fischeri," in *Proceedings of the 40th IEEE Conference on Decision and Control*, December 2001.

[6] A. A. Julius, Á. M. Halász, M. S. Sakar, H. Rubin, V. Kumar, and G. J. Pappas, "Stochastic modeling and control of biological systems: The lactose regulation system of escherichia coli," *IEEE Trans. Automat. Contr.*, vol. 53, pp. 51–65, 2008.

[7] S. Sankaranarayanan and G. Fainekos, "Simulating insulin infusion pump risks by in-silico modeling of the insulin-glucose regulatory system," in *International Conference on Computational Methods in Systems Biology*, 2012, [To Appear].

[8] G. Fainekos, S. Sankaranarayanan, K. Ueda, and H. Yazarel, "Verification of automotive control applications using s-taliro," in *Proceedings of the American Control Conference*, 2012.

[9] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Falsification of ltl safety properties in hybrid systems," in *Proc. of the Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, ser. LNCS, vol. 5505, 2009, pp. 368 – 382.

[10] T. Nghiem, S. Sankaranarayanan, G. Fainekos, F. Ivancic, A. Gupta, and G. Pappas, "Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems," in *Hybrid Systems: Computation and Control*, 2010.

[11] P. Zuliani, A. Platzer, and E. M. Clarke, "Bayesian statistical model checking with application to simulink/stateflow verification," in *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control*, 2010, pp. 243–252.

[12] G. E. Fainekos and G. J. Pappas, "Mtl robust testing and verification for lpv systems," in *Proceedings of the American Control Conference*, 2009, pp. 3748–3753.

[13] A. Rizk, G. Batt, F. Fages, and S. Soliman, "Continuous valuations of temporal logic specifications with applications to parameter optimization and robustness measures," *Theor. Comput. Sci.*, vol. 412, no. 26, pp. 2827–2839, 2011.

[14] A. Donze, G. Clermont, and C. J. Langmead, "Parameter synthesis in nonlinear dynamical systems: Application to systems biology," *Journal of Computational Biology*, vol. 17, no. 3, pp. 325–336, 2010.

[15] H. Abbas, G. E. Fainekos, S. Sankaranarayanan, F. Ivancic, A. Gupta, and G. J. Pappas, "Probabilistic temporal logic falsification of cyber-physical systems," *ACM Transactions on Embedded Computing Systems*, vol. (Accepted), 2011.

[16] G. Fainekos and G. Pappas, "Robustness of temporal logic specifications for continuous-time signals," *Theoretical Computer Science*, vol. 410, no. 42, pp. 4262–4291, September 2009.

[17] H. Abbas and G. Fainekos, "Linear hybrid system falsification through local search," in *Automated Technology for Verification and Analysis*, ser. LNCS, vol. 6996. Springer, 2011, pp. 503–510.

[18] R. Koymans, "Specifying real-time properties with metric temporal logic." *Real-Time Systems*, vol. 2, no. 4, pp. 255–299, 1990.

[19] M. M. Makela and P. Neittaanmaki, *Nonsmooth optimization*. World Scientific, 1992.

[20] S. Karaman, R. Sanfelice, and E. Frazzoli, "Optimal control of mixed logical dynamical systems with linear temporal logic specifications," in *IEEE Conf. on Decision and Control*, 2008.

[21] A. Donze and O. Maler, "Systematic simulation using sensitivity analysis," in *Hybrid Systems: Computation and Control*, ser. LNCS, vol. 4416. Springer, 2007, pp. 174–189.

[22] R. Serban and A. Hindmarsh, "Cvodes: the sensitivity-enabled ode solver in sundials," in *Proceedings of IDETC/CIE*, 2005.

[23] I. Hiskens and M. Pai, "Trajectory sensitivity analysis of hybrid systems," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 47, no. 2, pp. 204 –220, feb 2000.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[25] J. Goffin, "On convergence rates of subgradient optimization methods," *Mathematical Programming*, no. 13, pp. 329–347, 1977.