Convergence Proofs for Simulated Annealing Falsification of Safety Properties*

Houssam Abbas¹ and Georgios Fainekos²

Abstract—The problem of falsifying temporal logic properties of hybrid automata can be posed as a minimization problem by utilizing quantitative semantics for temporal logics. Previous work has used a variation of Simulated Annealing (SA) to solve the problem. While SA is known to converge to the global minimum of a continuous objective function over a closed and bounded search space, or when the search space is discrete, there do not exist convergence proofs for the cases addressed in that previous work. Namely, when the objective function is discontinuous, and when the objective is a vector-valued function. In this paper, we derive conditions and we prove convergence of SA to a global minimum in both scenarios. We also consider matters affecting the practical performance of SA.

I. INTRODUCTION

One of the major challenges in the model-based development of Cyber-Physical Systems (CPS) is how to automatically verify the correctness of a CPS model with respect to some formal specification. The proliferation of embedded computers in a multitude of safety critical systems and the well documented cases of CPS system failures due to software-physical system interactions [1], [2] demonstrate the urgency and the importance of the problem. However, it is well known [3] that the verification problem of CPS is undecidable, in general. Therefore, a lot of research effort has been focused on testing-based methodologies [4]–[13].

In previous work [14]–[16], a notion of robustness of temporal logics [17] is utilized as a cost function in order to convert the temporal logic falsification problem of CPS into an optimization problem. In detail, the robust semantics of a temporal logic formula over a CPS trajectory evaluate to a positive value if the trajectory satisfies the specification and to a negative value otherwise. Thus, we can convert the falsification problem into a minimization problem of the specification robustness over the set of all system trajectories. In general, the resulting optimization problem is non-convex and non-linear and the search space is uncountable. Thus, in [14]–[16], a number of stochastic heuristic search techniques were employed in order to solve the minimization problem with very promising results. In particular, in [14], a version of Simulated Annealing (SA) [18], [19] was utilized and a new SA heuristic for the minimization of a particular class

of vector functions was also proposed. For the former, it was claimed that under the assumption of a finite search space¹ we can guarantee convergence of SA to the global minimum and, thus, guarantee the solution of the original falsification problem. The convergence of the SA algorithm for vector functions was left as an open problem.

In this paper, we lift the assumption of the finite search space and we answer the question of what classes of CPS and under what conditions we can guarantee the convergence of SA to the global minimum. Furthermore, we derive conditions so that the SA algorithm over vector functions converges to the global minimum, as well. In brief, we prove that if a CPS is simulatable [20], then the SA algorithms are guaranteed to converge to the global minimum if the global minimum does not belong to an equivalence class of measure zero. The results in this paper are important because they help us understand the practical and theoretical limitations of the application of SA to the falsification problem of CPS. While they are presented for safety requirements due to space constraints, they can easily be extended to general temporal logic formulae.

Notation. $\mu(\cdot)$ denotes the Lebesgue measure. $\|\cdot\|$ is the Euclidean distance, and $B_\epsilon(x) = \{y \in \mathbb{R}^d \mid \|y - x\| < \epsilon\}$. $\mathbb{R}_{\geq 0} = [0, \infty)$ and $\mathbb{N}_{\geq 0} = \mathbb{R}_{\geq 0} \cap \mathbb{N}$. ‘Discrete set’ will mean a finite or countably infinite set. For a set X , $\mathcal{P}(X)$ is the set of all subsets of X .

II. PROBLEM FORMULATION

A. Falsification of safety properties of hybrid automata

We now introduce the practical setting in which the above two algorithms are applied, and all theorems are proven. We stress that we prove the correctness of the SA algorithms *when applied to this practical problem*. We consider a deterministic, non-Zeno² hybrid automaton [21]

$$\mathcal{H} = (L, X, Flow, Init, Inv, E, G, Re)$$

where $L = \{\ell_0, \ell_1, \dots\} \subset \mathbb{N}$ is a countable set of ‘locations’, $X \subset \mathbb{R}^d$ is the continuous state space, $Flow : L \times X \rightarrow X$ is a vector field describing the continuous evolution of a trajectory at (ℓ, x) , $Init \subset X$ is the set of initial continuous conditions, $Inv : L \rightarrow \mathcal{P}(X)$ associates an invariant set of $Flow(\ell, \cdot)$ to each location ℓ , $E \subset L \times L$ describes the possible jumps between locations (i.e. $(\ell_i, \ell_j) \in E$ iff there

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

¹H. Abbas is with the Department of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA hyabbas at asu.edu

²G. Fainekos is with the Department of Computing, Informatics and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA fainekos at asu.edu

¹Any compact set of initial conditions and/or other search parameters is going to be discretized to a finite set of floating-point numbers.

²An automaton is *Zeno* if it has trajectories that perform an infinite number of discrete jumps in a finite amount of time. This is an artifact of the modeling, and can not happen in reality.

exists a trajectory of the system that visits ℓ_i then ℓ_j without visiting any other location in-between), $G : E \rightarrow \mathcal{P}(X)$ defines guard conditions that cause jumps, and $Re : E \times X \rightarrow X$ is the reset map which resets the continuous state with every jump. $H = L \times X$ is the ‘state-space’ of $\mathcal{H}$. In this paper we consider automata for which $Init \subset Inv(\ell_0)$ so the initial set $H_0 = \{\ell_0\} \times Init$.

Given an initial state $h = (\ell_0, x) \in H_0$, the *hybrid trajectory* that starts there is a vector function $\eta_h : [0, \infty) \rightarrow H$ which associates a pair (location, continuous state) to each point in time: $\eta_h(t) = (\ell(t), s(x, t))$ where $s(x, t)$ is the continuous state at time t and $\ell(t)$ is the location of $s(x, t)$. $loc(h) = (\ell_0, \ell_1, \dots)$ is the list of locations visited by η_h , with no repetitions. At the time of the j^{th} jump time t_j , say between locations i and k , $\eta_h(t_j)$ is actually set-valued: $\eta_h(t_j) = \{(i, t_j), (k, t_j)\}$. Because there is no ambiguity about ℓ_0 , $\eta_{(\ell_0, x)}$ and $loc(\ell_0, x)$ will also be denoted by η_x and $loc(x)$ respectively.

We are given a safety property ϕ of the automaton, and the set $\mathcal{U} \subseteq X$ of states that violate ϕ . To falsify ϕ means to find an initial state $h_0 \in H_0$ such that the trajectory η_{h_0} that starts there enters the unsafe set $\mathcal{U}$. Here we only consider trajectories of finite duration $D < \infty$. With non-Zenoness, this implies that $loc(x)$ is always finite, even if L is countably infinite. L models the discrete variables of the automaton so the range of each x_i is *not* a discrete set³. We will need the following definitions [14]:

Definition 1 (Discrete distances): $\ell_{\mathcal{U}}$ is the location of the unsafe set $\mathcal{U}$. Let $G = (L, E)$ be the graph with vertex set L and edge set E . The *discrete distance* $\pi(\ell, \ell')$ between the two locations ℓ and ℓ' is the length of the shortest path in G between the locations. Given $x \in Init$ and the trajectory η_x , $\ell(x)$ is the location visited by η_x that is closest to $\ell_{\mathcal{U}}$, and $k(x)$ the corresponding distance:

$$\ell(x) = \operatorname{argmin}_{\ell \in loc(x)} \pi(\ell, \ell_{\mathcal{U}})$$

$$k(x) = \min_{\ell \in loc(x)} \pi(\ell, \ell_{\mathcal{U}}) = \pi(\ell(x), \ell_{\mathcal{U}})$$

Definition 2 (Continuous distances): For $x \in X$, $d_{\mathcal{U}}(x) = \inf_{y \in \mathcal{U}} \|x - y\|$ is the distance between x and the set $\mathcal{U}$. Given $(\ell, x) \in H$, $r(x) = \min_{t \leq D} d_Z(s(x, t))$, where Z is either $\mathcal{U}$ if $\ell_{\mathcal{U}} \in loc(x)$ (trajectory enters the location of $\mathcal{U}$), otherwise it is the guard leading to ℓ' , where ℓ' is the next location in a shortest path in G from ℓ to $\ell_{\mathcal{U}}$.

Definition 3 (Robustness): Given a trajectory η_x , its *robustness* is $V(x) \triangleq (k(x), r(x))$. This is the smallest ‘distance’ between the trajectory and the unsafe set $\mathcal{U}$ (note this isn’t a distance in the mathematical sense of the word). The *robustness time* $t(x)$ is the time when trajectory η_x is closest to $\mathcal{U}$: $\ell(x) = \ell(t(x))$ and $r(x) = d_Z(s(x, t(x)))$. Finding an x_0 that produces an unsafe trajectory (one that enters $\mathcal{U}$) can be achieved by finding the automaton’s trajectory with smallest robustness. Because only one initial location is possible (namely, ℓ_0), the search for a minimum

³If x_i is a discrete state variable taking values in $Q_i \subset \mathbb{Z}$, then L can be augmented to $L' = L \times Q_i$: every jump in the value of x_i can then be modeled as a change of locations in the extended set L' . The continuous dynamics are unaffected.

robustness-trajectory is to be carried over the initial set $Init$. The minimization problems treated in this paper are then:

Problem 1: For a dynamical system (with only one location, $L = \{\ell_0\}$)

$$\mathcal{D} = Init, \mathcal{R} = \mathbb{R}$$

$$V(x) = \min_{t \leq D} d_{\mathcal{U}}(s(t; x)) \quad (1)$$

$$\min_{x \in \mathcal{D}} V(x)$$

Problem 2: For a hybrid automaton:

$$\mathcal{D} = Init, \mathcal{R} = \mathbb{Z} \times \mathbb{R}$$

$$V(x) = (k(x), r(x)) \quad (2)$$

$$\min_{x \in \mathcal{D}} V(x)$$

B. Simulators of hybrid automata

Simulated Annealing (which will be formally defined in the next section) requires the ability to evaluate V at any point of the search space: this evaluation requires the simulation of a system trajectory starting at that point. A hybrid system simulator $\mathcal{H}_s$, which is necessarily a discretization of the real system $\mathcal{H}$, must be *accurate*, in the sense that for every simulated trajectory (generated by the simulator) starting at some $x_s \in Init_s$, there is an arbitrarily close real system trajectory (generated by $\mathcal{H}$) starting at $x \in Init$ ($Init_s$ is the discretization of $Init$).

Not every hybrid system admits an accurate simulator. In [20] sufficient conditions are given on $\mathcal{H}$ and $\mathcal{H}_s$ for $\mathcal{H}_s$ to be an accurate simulator of $\mathcal{H}$. The details of these conditions are given in the appendix. The following is a direct consequence of [20, Theorem 3.4]:

Proposition 1: Let $\mathcal{H}$ be a hybrid automaton, and let P be the partition of $Init$ induced by the equivalence relation $x \equiv x'$ iff $loc(x) = loc(x')$. Let $S \in P$ be a part such that $\mu(S) \neq 0$. If $\mathcal{H}$ satisfies the conditions for accurate simulation over S , then for any $x_0 \in S$ and every $\epsilon > 0$, there exists $\delta > 0$ with the following property: for every trajectory $\eta_x(\cdot)$ with initial point x in $B_\delta(x_0) \cap S$, and every $t < D$, there exists t' such that $|t - t'| < \epsilon$, $s(x, t')$ and $s(x_0, t)$ are in the same location, and $\|s(x, t') - s(x_0, t)\| < \epsilon$.

III. SIMULATED ANNEALING

Simulated Annealing (SA) is a well-known iterative stochastic algorithm for global optimization. We are given an objective function V with domain $\mathcal{D} \subset \mathbb{R}^d$ and range $\mathcal{R}$. $\mathcal{D}$ is known as the *state space*. The objective is to minimize V .

We are given a Markov kernel $R(\cdot, \cdot)$ on $(\mathcal{D}, \mathcal{B})$ where $\mathcal{B}$ is the Borel sigma field on $\mathcal{D}$. This is called the *transition* Markov kernel. Thus for each x in $\mathcal{D}$, $R(x, \cdot)$ is a probability measure on $(\mathcal{D}, \mathcal{B})$, and for each $B \in \mathcal{B}$, $R(\cdot, B)$ is a measurable function. We are also given a *cooling schedule* $(\tau_0, \tau_1, \dots)$ on $(\mathcal{D}, \mathcal{B})$: this is a sequence of (possibly random) positive numbers.

A. Traditional SA

In traditional SA, $\mathcal{D} \subset \mathbb{R}^d$ and $\mathcal{R} \subset \mathbb{R}$. SA constructs iteratively a sequence of *states* $(x_i) \in \mathcal{D}$, a sequence of *candidate points* $(y_i) \in \mathcal{D}$, and a sequence of *temperatures* $(\tau_i) \in \mathbb{R}_{>0}$, as follows: an initial state x_0 and an initial temperature τ_0 are given. Having constructed the sequences $(x_1, x_2, \dots, x_k)$, $(y_1, y_2, \dots, y_k)$, and $(\tau_1, \tau_2, \dots, \tau_k)$, a next candidate point y_{k+1} is selected according to the probability distribution $R(x_k, \cdot)$. The next state x_{k+1} is set

$$x_{k+1} = \begin{cases} y_{k+1} & \text{with probability } p(x_k, y_{k+1}, \tau_k) \\ x_k & \text{with probability } 1 - p(x_k, y_{k+1}, \tau_k) \end{cases}$$

where

$$p(x, y, \tau) = \min\{1, \exp[-\frac{V(y) - V(x)}{\tau}]\} \quad (3)$$

$p(x, y, \tau)$ is referred to as the *acceptance probability*. Note that if $V(y) < V(x)$ (so the candidate improves the value of the objective function), the candidate is accepted with certainty.

The following conditions are used in [18] to prove convergence:

- C0. The objective function V is continuous on $\mathcal{D}$.
- C1. The state space $\mathcal{D}$ is a bounded closed subset of $\mathbb{R}^d$.
- C2. There exists an $x_* \in \mathcal{D}$ such that V achieves its minimum at x_* .
- C3. Let x_* be as in C2. For every $\epsilon > 0$, the set $\{x \in \mathcal{D} \mid \|x - x_*\| < \epsilon\} = B_\epsilon(x_*) \cap \mathcal{D}$ has positive Lebesgue measure.
- C4. The selection Markov kernel R is absolutely continuous (with respect to the Lebesgue measure on $\mathbb{R}^d$) and it has a density which is uniformly bounded away from 0. That is, R is of the form

$$R(x, B) = \int_B r(x, y) dy \text{ with } \inf_{x, y \in \mathcal{D}} r(x, y) > 0$$

This implies that all of $\mathcal{D}$ is reachable from any $x \in \mathcal{D}$.

- C5. For every open subset B in $\mathcal{D}$, $R(x, B)$ is continuous in x .
- C6. For every choice of initial state x_0 and initial temperature τ_0 , the sequence of temperatures $(\tau_k)_{k \geq 0}$ converges in probability to 0.

The following theorem is proven in [18]:

Theorem 1: Let $x_1, x_2, x_3, \dots$ be the sequence of states generated by the SA algorithm with selection Markov kernel R and with cooling schedule τ . Assume that conditions C0-C6 hold. Let V_* denote the global minimum of V on $\mathcal{D}$. Then, for every choice of initial conditions (x_0, τ_0) , the sequence of function values $(V(x_k))_{k \geq 0}$ converges in probability to V_* . That is, $\forall \epsilon > 0, \Pr[|V(x_k) - V_*| > \epsilon] \rightarrow 0$ as $k \rightarrow \infty$.

We now discuss the applicability of conditions C0-C6 to our hybrid automata:

C0: C0 does not hold for Problem 2, and one contribution of this paper is to show convergence in the absence of global continuity. Another contribution is to show that it holds for

Problem 1.

C1: The initial state $x_0 \in \text{Init}$ typically models starting physical parameters of the system, and these are always finite in magnitude, whence Init is bounded. Closure of Init will have to be assumed (this is a standard assumption, in both practice and theory [4], [20], [22], [23]).

C2: It is shown in Section IV that C0 and C1 imply C2, so it holds for Problem 1. We assume it holds for Problem 2.

C3: We first generalize C3 to hybrid automata in condition C7:

- C7. Let x_* be as in C2, and let $S_* \in P$ be the part to which it belongs. Then $\mu(B_\delta(x_*) \cap S_*) > 0 \forall \delta > 0$.

It is immediate to see that the probability of converging to a minimum that does not satisfy C7 (such as an isolated point) is 0. Therefore our results can only claim convergence to minima that belong to sets of non-zero measure, and this is captured in C7.

C4-C6: these are properties of the optimization algorithm rather than of the system. The Hit-and-Run sampler may be used to satisfy C4 and C5 [14], and the cooling schedule can be chosen to satisfy C6.

We introduce one more condition on the automaton:

- C8. The hybrid automaton admits an accurate simulator over Init .

Without this condition, it is not guaranteed that we can draw conclusions about the real system, based on simulations.

B. SA for minimizing a vector function

We now consider the case $\mathcal{R} = \mathbb{Z} \times \mathbb{R}$, so $V(x) = (V_1(x), V_2(x))$ is a vector function. The range of V is lexicographically ordered: $(k, r) \leq (k', r')$ iff $(k < k')$ OR $(k = k' \text{ and } r \leq r')$. This is a total order, so the issue of non-dominance [24] does not arise. The resulting *vector SA* algorithm, introduced in [14], constructs a sequence of states $(x_i) \in \mathcal{D}$, a sequence of candidate points $(y_i) \in \mathcal{D}$, and a sequence of temperatures $(\tau_i) \in \mathbb{R}_{>0}$, as follows: an initial state $x_0 \in \mathcal{D}$ and an initial temperature $\tau_0 > 0$ are given. Having constructed the sequences $(x_1, x_2, \dots, x_k)$, $(y_1, y_2, \dots, y_k)$, and $(\tau_1, \tau_2, \dots, \tau_k)$, a next candidate point y_{k+1} is selected according to the probability distribution $R(x_k, \cdot)$. It then computes

$$\begin{aligned} \alpha_i &= \exp\left[-\frac{V_i(y_{k+1}) - V_i(x_k)}{\tau_k}\right], \quad i = 1, 2 \\ u &= \text{UniformRandomReal}(0, 1) \end{aligned}$$

The next state x_{k+1} is determined as

- $x_{k+1} = y_{k+1}$ if the event $A = (V_1(x_k) = V_1(y_{k+1}) \wedge u \leq \alpha_2) \vee (V_1(x_k) \neq V_1(y_{k+1}) \wedge u \leq \alpha_1)$ is true
- and $x_{k+1} = x_k$ otherwise.

The two events on either side of the disjunctive are disjoint. It comes that the update rule for vector SA is:

$$x_{k+1} = \begin{cases} y_{k+1} & \text{with probability } p_a(x_k, y_{k+1}, \tau_k) \\ x_k & \text{with probability } 1 - p_a(x_k, y_{k+1}, \tau_k) \end{cases}$$

where

$$p_a(x, y, \tau) = \min\{1, \sum_{i=1}^2 \Pr[V_i(y_{k+1}) \neq V_i(x_k) | u \leq \alpha_{3-i}] \cdot \alpha_{3-i}\}$$

We conclude with a Lemma, used in the later proofs, which will allow us to forego continuity of V over $Init$. It is proved for vector V , which subsumes scalar V as a special case. The global minimum of $V_* = (k_*, r_*)$ of V will be characterized by

$$\begin{aligned} k_* &\leq k(x) \quad \forall x \in \mathcal{D} \\ r_* &\leq r(x) \quad \forall x \in \{y \in \mathcal{D} | V_1(y) = k_*\} \end{aligned}$$

Given $\epsilon = (\epsilon_1, \epsilon_2) \in \mathbb{N}_{>0} \times \mathbb{R}_{>0}$, define

$$\begin{aligned} \mathcal{D}_\epsilon &= \{x \in \mathcal{D} | V_1(x) \leq k_* + \epsilon_1 \text{ and } V_2(x) \leq r_* + \epsilon_2\} \\ &= \{x \in \mathcal{D} | V(x) \leq V_* + \epsilon\} \end{aligned}$$

Lemma 1: Assume C7, C8. Then it holds that $\forall \epsilon = (\epsilon_1, \epsilon_2) > 0$, $0 < \mu(\mathcal{D}_\epsilon)$.

Proof: Fix $\epsilon = (\epsilon_1, \epsilon_2) > 0$, and consider the optimum x_* and its robustness time $t_* = t(x_*)$. By C8 and Prop.1, $\exists \delta_2 > 0$ s.t. the image of $B_{\delta_2}(x_*) \cap S_* \triangleq B_{\delta_2}$ under the hybrid dynamics at time t_* is a set of points that are at least ϵ_2 -close to $s(x_*, t_*)$. Moreover, all trajectories η_y starting in B_{δ_2} follow the same sequence of locations as η_{x_*} ; in particular, they all visit the location $l(x_*)$ and therefore have $k(y) = k(x_*)$. So $|k(y) - k(x_*)| < \epsilon_1$. We know that $\mu(B_{\delta_2}) > 0$ by condition C7. Recognizing that $B_{\delta_2} \subset \mathcal{D}_\epsilon$ proves the lemma. ■

Optimizing a vector function. One popular way to optimize a vector objective function is to map its output $V(x) = (k, r)$ to scalars in $\mathbb{R}$ (e.g. [25]), e.g. using the inverse logit function with $a > 0$

$$Y_a : (k, r) \mapsto k + 2[2 \exp(r/a)/(1 + \exp(r/a)) - 1]$$

Y_a maps $\mathbb{Z} \times \mathbb{R}$ to $\{z + b \mid z \in \mathbb{Z}, b \in (-1, 1)\}$. It is strictly increasing, so the global minima of $Y_a \circ V$ are the same as the global minima of V . However, it faces the ‘saturation’ effect for large absolute values of r (and fixed k): differences between Y_a values become insignificant at these extremes, thus not providing the optimizer with enough guidance. This problem is exacerbated in a practical implementation, which will discretize Y_a , because the the discretized inverse logit is no longer strictly increasing. This means the global minima of $\tilde{Y}_a \circ V$ are no longer the same as those of V .

In section IV, it is shown that traditional SA converges when solving Problem 1. In section V, it is shown that vector SA converges when solving Problem 2. Throughout, any mention of a global minimum refers to a minimum that satisfies all needed conditions, which will be explicated.

IV. TRADITIONAL SA FOR A DYNAMICAL SYSTEM

Our first result states that traditional SA converges to the global minimum V_* when solving Problem 1. Recall that Problem 1 deals with automata with one location, so there are no guards or resets. This result may be seen as a

special case of Thm.3 presented in the next section. But it is presented here as a separate result, as a first simple extension of SA convergence, and to highlight the role played by the condition that $\mu(\mathcal{D}_\epsilon) > 0$ for all $\epsilon > 0$.

Theorem 2: Assume that conditions C1,C2,C4-C8 are satisfied. Then SA will converge in probability to the global minimum of Problem 1.

Proof: The proof proceeds along identical lines to Belisle’s original proof in [18] and so is not repeated. The main difference is that V is not continuous over its domain $Init$ in the present paper. However, Lemma 1 removes the need for this ‘global’ continuity of the objective function, and may be invoked when proving [18, Lemma 1]. ■

V. VECTOR OBJECTIVE FUNCTION

In this section we prove the convergence of vector SA when solving Problem 2. The novelty is that this SA algorithm deals with a multi-objective function $V(x) = (k, r) \in \mathbb{N} \times \mathbb{R}$, and the objective is no longer continuous on $\mathcal{D}$.

The proof is a variation on Belisle’s proof [18]. Some of the definitions need to be appropriately modified to account for the new objective function. The following theorem is proven next.

Theorem 3: Let $x_1, x_2, \dots$ be the sequence of states generated by vector SA when solving Problem 2. Assume that conditions C1,C2,C4-C8 hold. Let V_* denote the global minimum of V on $\mathcal{D}$. Then, for any pair of initial conditions (x_0, τ_0) , the sequence of function values $(V(x_k)), k \geq 0$, converges in probability to V_* .

Since V is a vector function, all expressions are understood to apply in a component-wise fashion, e.g. $|V(x)| > 0$ iff $|V_1(x)| > 0$ and $|V_2(x)| > 0$.

We will show that for every $x_0 \in \mathcal{D}, t_0 > 0, \epsilon > 0$ and $\delta > 0$ there exists an integer n_1 such that

$$\Pr[X_n \notin \mathcal{D}_\epsilon | (X_0, T_0) = (x_0, \tau_0)] \leq \delta \quad \forall n \geq n_1 \quad (4)$$

This will prove the theorem. If ϵ is such that $Init \subseteq \mathcal{D}_\epsilon$ then Eq.(4) is trivially satisfied. Therefore, fix $\epsilon > 0$ such that $\mathcal{D}_\epsilon \subset Init$, $x_0 \in \mathcal{D}, \tau_0 > 0$ and $\delta > 0$. Let m and n be positive integers, $\zeta \in \mathbb{N}_{>0} \times \mathbb{R}_{>0}$, and $X_n^m = (X_n, X_{n+1}, \dots, X_m)$ be the random sequence generated by the random process $\{X_n\}$. x_n^m will be a realization of X_n^m . Define the following events:

$A = A(m, n)$ = the event that none of the states $(X_i)_{i=n}^{n+m}$ is in $\mathcal{D}_\epsilon$

$B = B(\zeta, m, n)$ = the event that at least one of the transitions $X_{n+(k-1)} \rightarrow X_{n+k}$, $k = 0, 1, \dots, m$, is a move from $\mathcal{D}_\epsilon$ to $H_{\epsilon, \zeta} \triangleq \{x \in \mathcal{D} | V_* + \epsilon < V(x) \leq V_* + \epsilon + \zeta\}$.

$C = C(\zeta, m, n)$ = the event that at least one of the transitions $X_{n+(k-1)} \rightarrow X_{n+k}$, $k = 0, 1, \dots, m$, is a move from $\mathcal{D}_\epsilon$ to $\tilde{H}_{\epsilon, \zeta} \triangleq \{x \in \mathcal{D} | V_* + \zeta + \epsilon < V(x)\}$.

D = the event that $X_{n+m} \notin \mathcal{D}_\epsilon$.

Observe that $D \subset A \cup B \cup C$. Thus for every ζ, m and n

we have

$$\begin{aligned}
& \Pr[X_{n+m} \notin \mathcal{D}_\epsilon | (x_0, \tau_0)] \\
&= \Pr[D | (x_0, \tau_0)] \\
&\leq \Pr[A | (x_0, \tau_0)] + \Pr[B | (x_0, \tau_0)] \\
&+ \Pr[D | (x_0, \tau_0)]
\end{aligned} \tag{5}$$

In the next three sections we prove the following three lemmata, which show that the ‘escape’ probabilities on the right hand side of (5) can be made arbitrarily small, regardless of initial conditions (x_0, τ_0) .

Lemma 2: There exists an integer m_0 (which does not depend on (x_0, τ_0)), such that $\Pr[A(m_0, n) | (x_0, \tau_0)] < \delta/3 \forall n \geq 0$

Lemma 3: Let m_0 be as in Lemma 2. There exists a $\zeta_0 = (\zeta_{0,1}, \zeta_{0,2})$, independent of (x_0, τ_0) , such that

$$\Pr[B(\zeta_0, m_0, n) | (x_0, \tau_0)] < \delta/3 \forall n \geq 0$$

Lemma 4: Let m_0 and ζ_0 be as in Lemma 2 and Lemma 3 resp. There exists an integer n_0 , independent of (x_0, τ_0) , such that

$$\Pr[C(\zeta_0, m_0, n) | (x_0, \tau_0)] < \delta/3 \forall n \geq n_0$$

Thus we may conclude that $\Pr[X_{n+m} \notin \mathcal{D}_\epsilon | (x_0, \tau_0)] < \delta \forall n \geq n_0$. Therefore, Eq.(4) holds with $n_1 = n_0 + m_0$.

The proof of Lemma 2 is almost identical to the original proof in [18] with some minor obvious modifications, and again, uses Lemma 1 instead of continuity of V . Therefore it is omitted.

A. Proof of Lemma 3

Let $X_i \rightarrow X_{i+1}$ denote a transition from state X_i to state X_{i+1} , and $\zeta \in \mathbb{N}_+ \times \mathbb{R}_+$.

$$\begin{aligned}
& \Pr[B(\zeta, m_0, n) | (x_0, \tau_0)] \\
&\leq \sum_{j=0}^{m-1} \Pr[\underbrace{X_{n+j} \in \mathcal{D}_\epsilon \rightarrow X_{n+j+1} \in H_{\epsilon, \zeta}}_{B_j(\zeta, m_0, n)} | (x_0, \tau_0)] \\
&\Pr[B_j(\zeta, m_0, n) | (x_0, \tau_0)] \leq \sup_{x \in \mathcal{D}} R(x, H_{\epsilon, \zeta})
\end{aligned}$$

Thus

$$\Pr[B(\zeta, m_0, n) | (x_0, \tau_0)] \leq m_0 \sup_{x \in \mathcal{D}} R(x, H_{\epsilon, \zeta}) \tag{6}$$

Now we may write

$$\begin{aligned}
H_{\epsilon, \zeta} &= \{x \in \mathcal{D} \mid V_{1*} + \epsilon_1 < V_1(x) \leq V_{1*} + \epsilon_1 + \zeta_1\} \\
&\cap \{x \in \mathcal{D} \mid V_{2*} + \epsilon_2 < V_2(x) \leq V_{2*} + \epsilon_2 + \zeta_2\} \\
&\triangleq H_{\epsilon, \zeta}^1 \cap H_{\epsilon, \zeta}^2
\end{aligned}$$

Consider a sequence $\{\zeta^{(i)}\}$ s.t. $\zeta^{(i)} \rightarrow 0$ as $i \rightarrow \infty$. Then $\mathcal{D} \supseteq H_{\epsilon, \zeta^{(1)}}^1 \supseteq H_{\epsilon, \zeta^{(2)}}^1 \supseteq \dots$: since $\mathcal{D}$ is bounded (C1), $H_{\epsilon, \zeta}^1$ is bounded and therefore has finite Lebesgue measure. Next, $\mu(H_{\epsilon, \zeta}^2) \rightarrow 0$ as $\zeta_2 \rightarrow 0$ (Lemma 2 in [18] with $\zeta_2 = 1/\ell$). Thus $\mu(H_{\epsilon, \zeta}^1 \cap H_{\epsilon, \zeta}^2) = \mu(H_{\epsilon, \zeta}) \rightarrow 0$ as $\zeta \rightarrow 0$. Therefore, for every $x \in \mathcal{D}$, $R(x, H_{\epsilon, \zeta}) = \int_{x \in H_{\epsilon, \zeta}} r(x, y) dy \rightarrow 0$ as $\zeta \rightarrow 0$. Thus:

- $R(x, H_{\epsilon, \zeta^{(i)}})$ is a real-valued continuous function over a compact space $\mathcal{D} \forall i$.

- $\{R(x, H_{\epsilon, \zeta^{(i)}})\}_i$ is a monotonically decreasing sequence of functions.
- The sequence $\{R(x, H_{\epsilon, \zeta^{(i)}})\}$ converges pointwise to the 0 function.

Dini’s theorem allows us to conclude that $R(x, H_{\epsilon, \zeta}) \rightarrow 0$ as $\zeta \rightarrow 0$ uniformly in x . Thus $\exists \zeta_0 = (\zeta_{1,0}, \zeta_{2,0})$ s.t. $\sup_{x \in \mathcal{D}} R(x, H_{\epsilon, \zeta_0}) < \delta/3m_0$. Combined with (6), this proves the lemma.

B. Proof of Lemma 4

Let m_0 and ζ_0 be as in Lemma 1 and Lemma 2 respectively. Fix $\tau_* > 0$.

$$\begin{aligned}
& \Pr[C(\zeta_0, m_0, n) | (x_0, \tau_0)] \\
&\leq \sum_{j=0}^{m-1} \Pr[\underbrace{X_{n+j} \in \mathcal{D}_\epsilon \rightarrow X_{n+j+1} \in \tilde{H}_{\epsilon, \zeta_0}}_{C_j(\zeta_0, m_0, n)} | (x_0, \tau_0)].
\end{aligned}$$

$$\begin{aligned}
& \Pr[C_j(\zeta_0, m_0, n) | (x_0, \tau_0)] \\
&= \Pr[C_j(\zeta_0, m_0, n) \cap \tau_{n+j} \leq \tau_* | (x_0, \tau_0)] \\
&+ \Pr[C_j(\zeta_0, m_0, n) \cap \tau_{n+j} > \tau_* | (x_0, \tau_0)]
\end{aligned}$$

The first summand is upper bounded by

$$\begin{aligned}
& p_a(x, y, \tau), x \in \mathcal{D}_\epsilon, y \in \tilde{H}_{\epsilon, \zeta_0}, \tau \leq \tau_* \\
&\leq \alpha_1 + \alpha_2 \\
&= \exp \left[-\frac{V_1(y) - V_1(x)}{\tau} \right] + \exp \left[-\frac{V_2(y) - V_2(x)}{\tau} \right] \\
&\leq \exp \left[\frac{-V_* - \epsilon_1 - \zeta_{1,0} + V_* + \epsilon_1}{\tau} \right] \\
&+ \exp \left[\frac{-V_* - \epsilon_2 - \zeta_{2,0} + V_* + \epsilon_2}{\tau} \right] \\
&\leq \exp \left[\frac{-\zeta_{1,0}}{\tau} \right] + \exp \left[\frac{-\zeta_{2,0}}{\tau} \right]
\end{aligned}$$

So there exists a τ_* s.t. $\Pr[C(\zeta_0, m_0, n) | (x_0, \tau_0)] \leq \delta/6$ when $\tau \leq \tau_*$. Condition C6 guarantees the existence of an n_2 such that $\tau_n \leq \tau_*$ for all $n \geq n_2$. The second summand can be made arbitrarily small by C6: in particular, let n_3 be such that it is smaller than $\delta/6$. Letting $n_0 \geq \max\{n_2, n_3\}$ proves Lemma 4.

VI. PRACTICAL CONSIDERATIONS

The previous sections have demonstrated that SA is a consistent optimization algorithm for the falsification of temporal logical properties of hybrid automata; that is, the sequence of samples it produces converges (in probability) to the set of global optima, regardless of the initial sample. This is an important property since it guarantees that longer runs of SA will produce better minima. From a practical standpoint, previous work [14] has demonstrated that in practice, SA performs well, both in terms of speed and quality of obtained minimum. Rather than replicate those experiments here, we focus instead on the factors that affect finite-time performance⁴, and how they affect it.

⁴Finite-time performance tells us how close is the current minimum, after N samples generated, to the global minimum. The answer is naturally affected by the likelihood of SA to spend many samples near non-global minima for this system and specification.

It is well-known that the practical performance of SA depends on the specific objective function being optimized, the particular cooling schedule, and the neighborhood selection; see [26] for a good review of these issues. In our case, the neighborhood is all of $Init$ as per condition C4, and the cooling schedule is adaptively modified to maintain an acceptance-to-rejection ratio close to 1. This is permitted by condition C6, and has been shown experimentally to help avoid local minima traps [14]. The objective function is directly related to the system and property being falsified, and we now briefly illustrate how its graph can affect convergence. We select two benchmark hybrid automata and corresponding unsafe sets, and study the following three issues for each:

- Generate the partition P of its $Init$ set. This allows us to assess whether it satisfies condition C7.
- Plot the graph of the objective function, which can indicate the difficulty of this problem instance.
- See if vector SA generates samples with different sequences of locations (ℓ_i) for our system. If a run of vector SA generates very few different (ℓ_i), this might indicate a local minimum trap, which should not be confused with having converged (indeed, Lemma 2 asserts that the tail of the generated sequence consists of samples with the same sequence of locations with increasing probability).

Our first system, Sys1, is a 2D, 5-location hybrid automaton with linear dynamics in each location:

$$\begin{aligned}
 Flow(1, x) &= 0.1 \begin{bmatrix} -1 & 10 \\ -100 & -1 \end{bmatrix} x, Flow(2, x) = 0.1 \begin{bmatrix} -1 & 100 \\ -10 & -1 \end{bmatrix} x \\
 Flow(3, x) &= Flow(4, x) = 0.1 \begin{bmatrix} -1 & 100 \\ -10 & -1 \end{bmatrix} x \\
 Flow(5, x) &= 0.1 \begin{bmatrix} 1 & -10 \\ 10 & 1 \end{bmatrix} x \\
 Guard(1, 2) &= \{x_1 > 0\}, Guard(2, 3) = \{x_2 < 0 \wedge x_1 < 4.5\} \\
 Guard(3, 4) &= \{x_1 < 0\}, Guard(2, 5) = \{x_2 < 0 \wedge x_1 > 4.5\} \\
 Guard(4, 1) &= \{x_2 > 0\}, Guard(5, 3) = \{x_1 < 4.5\} \\
 \mathcal{U} &= \{3 < x_1 < 4, 3 < x_2 < 4\}
 \end{aligned}$$

To generate the graph of the robustness function, we sampled the initial set with a step size of 0.01 in both dimensions, leading to a grid with 3600 points. The graph (not shown here for lack of space) displays several near-flat ‘valleys’ (regions of small function values) surrounded by ‘peaks’ (regions of large function values): if SA samples from a given valley, it will continue sampling from it for a long time because the probability of accepting an increase in function value will be small, following Eq.(3). Thus we expect that once SA samples from a valley that contains a global minimum, then there’s a high probability it will get arbitrarily close to that minimum.

Using the same grid, the partition P of the initial set was obtained. A coarser partition (with step size = 0.02) is shown in Fig.1 for clarity. A total of 4 parts were obtained, each corresponding to a sequence of locations. It can be seen that Parts 1 and 3 cover much of $Init$ (see Fig.1), so it is to be expected that at least initially, SA will sample from these 2 parts overwhelmingly. If the global minimum is in a different part, longer runs of SA are required.

Fig. 1: $Init$ partition for Sys1. Parts correspond to the sequences s_i

We then ran SA on Sys1 with a sample size of 1000 initial points, and a trajectory duration of 2sec. All 4 parts were visited, with the vast majority of the points chosen from the Part3. This is in accordance with the observations made above about the partition of $Init$ and shape of the graph. It is notable that the minimum found by SA (down to two significant figures) is $[1.35, 1.74]$, almost at the boundary of parts 3 and 4. These points are harder to find for SA because of the different sequences on either side of the boundary. The corresponding robustness value is 0.5 (compare to global minimum robustness value of 0.49, found on above grid, at $[1.58, 1.79]$).

We also re-use the Nav0 benchmark from [27], which we will argue is a harder instance for SA. Nav0 is a 4-dimensional automaton with 16 locations, and it is unknown whether it is falsifiable or not. For the purposes of presenting results graphically, we fix the last 2 dimensions of the state vector to $[x_3, x_4] = [0.1, 0.2]$, and let SA vary the first two dimensions. The graph of the robustness function is given in Fig.2. The graph was obtained by sampling the initial set with a step size of 0.01 (for a total of 3600 points), and computing a trajectory of duration 10sec. We can observe a large number of minima with varying depths, increasing the odds of SA spending a large number of samples in these minima before jumping back out.

Fig. 2: Graph of the robustness function for Nav0.

Using the same grid, the partition P of the initial set

was computed. The partition has 189 parts, with the largest part containing only 15.3% of the points, and the 4 largest parts together have 53.3% of the points. Fig.3 shows how the initial set of conditions is highly fragmented. While we haven't established a precise relation between the measure of the parts and SA convergence, condition C7 and the proof of Prop.1 drive us to conjecture that a larger measure for the parts leads to a faster convergence. The small size of all partitions in Nav0 then suggests this is a hard instance for SA.

Fig. 3: Zoom on *Init* partition for Nav0. Different symbols and colors correspond to different sequences of locations. Note the fragmentation on the right side.

SA was run on Nav0 with a sample size of 1000 initial points, and a test duration of 10sec. It selected points from 62 parts, with the largest 2 containing 70% of the 1000 points, and the smallest containing 0.01% of the points. These parts were *not* the largest parts in P , which is expected since SA is driven by the objective function as well as the measure of the parts. SA found a global minimum value of 0.09 at $[0.61, 3.40]$, compared to the global (grid) minimum of 0.08 at $[0.67, 3.52]$.

On the theoretical side, the probabilistic convergence of the sequence of initial states generated by SA to a global optimum, can be informally divided into two parts: the first part is the convergence of the Markov Chain to its zero-temperature stationary distribution, π_∞ , in an appropriate sense. The stationary distribution favors minima of the objective function. The second part is then the sampling from the stationary distribution. Thus a bound on SA's convergence involves bounding these two components. The second part depends on the shape of the stationary distribution. In our case, it is exponential of the form $\pi_\infty(x) = \frac{1}{M} e^{-\beta V(x)}$ [14], where M is the (unknown) normalization constant. For the first part, recently, a result on finite-time *guarantees* for SA optimization over continuous domains was obtained in [28]. Informally, given a desired precision of the optimization, it

provides a number of samples after which we are guaranteed that the minimum so far is within the desired precision of the global minimum. The strength of this result is that it requires very little of the objective function, namely only that it be well-defined pointwise, measurable, and bounded between 0 and 1. In traditional SA, $g \circ V(x)$ satisfies these conditions, with V given in Problem 1 and g is any monotone increasing function that maps $[0, \infty]$ to $[0, 1]$. No such result exists yet for a vector objective function, like the one in Problem 2; this is the subject of future research.

VII. CONCLUSIONS

The problem of falsifying safety properties of hybrid automata was formulated as optimization problems in [14]. In this paper, we provided conditions on the system under which Simulated Annealing will converge in probability to the global minimum, and thus return a system trajectory of minimal robustness. Research can proceed along three directions: the first is to establish convergence conditions for discrete implementations of the continuous models studied in this paper. A second direction is to broaden the class of systems for which SA converges, e.g. to systems with more than one starting location, and to automata with state- and time-dependent guard sets, as well as to non-autonomous systems. A third direction of research is to develop computable criteria that establish whether a given system is, a priori, suitable for efficient SA falsification or not. As part of this direction, there is obvious interest in establishing finite-time guarantees for the vector SA algorithm.

ACKNOWLEDGMENT

The authors wish to thank Tolga Duman for many helpful discussions.

REFERENCES

- [1] M. Blair, S. Obenski, and P. Bridickas, "Patriot missile software problem," United States General Accounting Office, Tech. Rep. GAO/IMTEC-92-26, 1992.
- [2] E. J. Hoffman, W. L. Ebert, M. D. Femiano, H. R. Freeman, C. J. Gay, C. P. Jones, P. J. Luers, and J. G. Palmer, "The near rendezvous burn anomaly of december 1998," Applied Physics Laboratory, Johns Hopkins University, Tech. Rep., Nov. 1999.
- [3] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?" *J. Comput. Syst. Sci.*, vol. 57, no. 1, pp. 94–124, 1998.
- [4] S. Ratschan and J.-G. Smaus, "Finding errors of hybrid systems by optimizing an abstraction-based quality estimate," in *Proceedings of the Third Int'l Conf. on Tests and Proofs*, Zurich, Switzerland, July 2009, pp. 153–168.
- [5] Q. Zhao, B. H. Krogh, and P. Hubbard, "Generating test inputs for embedded control systems," *IEEE Control Systems Magazine*, vol. Aug., pp. 49–57, 2003.
- [6] M. Branicky, M. Curtiss, J. Levine, and S. Morgan, "Sampling-based planning, control and verification of hybrid systems," *IEE Proc.-Control Theory Appl.*, vol. 153, no. 5, pp. 575–590, 2006.
- [7] T. Nahhal and T. Dang, "Test coverage for continuous and hybrid systems," in *CAV*, ser. LNCS, vol. 4590. Springer, 2007, pp. 449–462.
- [8] E. Plaku, L. E. Kavradi, and M. Y. Vardi, "Hybrid systems: From verification to falsification," in *Proceedings of the 19th International Conference on Computer Aided Verification*, ser. LNCS, W. Damm and H. Hermanns, Eds., vol. 4590. Springer, 2007, pp. 463–476.
- [9] —, "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] A. Rizk, G. Batt, F. Fages, and S. Soliman, "On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology," in *International Conference on Computational Methods in Systems Biology*, ser. LNCS, no. 5307. Springer, 2008, pp. 251–268.
- [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] 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.
- [13] F. Lerda, J. Kapinski, E. M. Clarke, and B. H. Krogh, "Verification of supervisory control software using state proximity and merging," in *Hybrid Systems: Computation and Control*, ser. LNCS, vol. 4981. Springer, 2008, pp. 344–357.
- [14] 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.
- [15] Y. S. R. Annappureddy, C. Liu, G. E. Fainekos, and S. Sankaranarayanan, "S-taliro: A tool for temporal logic falsification for hybrid systems," in *Tools and algorithms for the construction and analysis of systems*, ser. LNCS, vol. 6605. Springer, 2011, pp. 254–257.
- [16] 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.
- [17] 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.
- [18] C. J. P. Belisle, "Convergence theorems for a class of simulated annealing algorithms on $\mathbb{R}^d$," *Journal of Applied Probability*, vol. 29, no. 4, pp. 885–895, Dec. 1992.
- [19] B. Hajek, "Cooling schedules for optimal annealing," *Mathematics of operation research*, vol. 13, no. 2, pp. 311–329, 1988.
- [20] R. G. Sanfelice and A. R. Teel, "Dynamical properties of hybrid systems simulators," *Automatica*, vol. 46, no. 2, pp. 239–248, 2010.
- [21] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009.
- [22] A. Girard and G. J. Pappas, "Approximation metrics for discrete and continuous systems," *IEEE Trans. Auto. Cont.*, vol. 52, no. 5, pp. 782–798, 2007.
- [23] J. Lygeros, K. H. Johansson, S. N. Simic, J. Zhang, and S. Sastry, "Dynamical properties of hybrid automata," *IEEE Transactions on Automatic Control*, vol. 48, pp. 2–17, 2003.
- [24] K. I. Smith, R. M. Everson, J. E. Fieldsend, C. Murphy, and R. Misra, "Dominance-based multiobjective simulated annealing," *IEEE Transactions on Evolutionary computation*, vol. 12, no. 3, pp. 323–342, 2008.
- [25] P. Czyzak and A. Jaszkievicz, "Pareto simulated annealing - a meta-heuristic technique for multiple-objective combinatorial optimization," *J. Multi-Criteria Decision Analysis*, vol. 7, pp. 34–47, 1998.
- [26] D. Henderson, S. H. Jacobson, and A. W. Johnson, "The theory and practice of simulated annealing," in *Handbook of metaheuristics*. Springer, 2003.
- [27] 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.
- [28] A. Lecchini-Visintini, J. Lygeros, and J. M. Maciejowski, "Stochastic optimization on continuous domains with finite-time guarantees by markov chain monte carlo methods," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2858–2863, Dec. 2010.
- [29] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *IEEE Control Systems Magazine*, pp. 28–93, 2009.

VIII. APPENDIX

This section details the ‘hybrid basic conditions’ that are required of a hybrid automaton $\mathcal{H}$ for it to have an accurate simulator. The conditions are framed in the formalism of differential and difference inclusions [29], which generalizes the formalism used in this paper.

Definition 4: [20] A hybrid system $\mathcal{H}$ on a state space $\mathbb{R}^n$ is defined by

- A set $C \subseteq \mathbb{R}^d$ called the *flow set*
- A set-valued map $F : \mathbb{R}^n \rightarrow \mathcal{P}(\mathbb{R}^n)$ called the *flow map*
- A set $D \subseteq \mathbb{R}^n$ called the *jump set*
- A set-valued map $G : \mathbb{R}^n \rightarrow \mathcal{P}(\mathbb{R}^n)$ called the *jump map*

We write $\mathcal{H} = (C, F, D, G)$. The dynamics of the system are given by

$$\dot{\xi} \in \mathbb{R}^n \begin{cases} \dot{\xi} \in F(\xi), \xi \in C \\ \xi^+ \in G(\xi), \xi \in D \end{cases}$$

Hybrid automata, defined in section II-A, can be modeled using Def. 4 as follows [29]: let $n = d + 1$ and $\xi = [\ell, x]^T \in L \times \mathbb{R}^d \subset \mathbb{R}^{d+1}$ be the state of the system. Then we take: $C_\ell = \text{Inv}(\ell)$, $D_\ell = \cup_{\ell': (\ell, \ell') \in E} \text{Guard}(\ell, \ell')$, $F_\ell(x) = \text{Flow}(\ell, x)$, $\forall x \in C_\ell$, and $G_\ell(x) = \{[\ell', \text{Re}(\ell, \ell', x)]^T \mid x \in \text{Guard}(\ell, \ell')\}$, $\forall x \in D_\ell$.

For a deterministic automaton, for any x , there is a unique location ℓ' such that $x \in \text{Guard}(\ell, \ell')$ (otherwise, two jumps (ℓ, ℓ') and (ℓ, ℓ'') are possible). Thus $G_\ell(x)$ is a singleton. The dynamics of $\mathcal{H}$ are then given by:

$$[\ell, x]^T \in L \times \mathbb{R}^d \begin{cases} \dot{\xi} \in F(\xi) = \{[0, F_\ell(x)]^T\}, x \in C_\ell \\ \xi^+ \in G(\xi) = \{[\ell', \text{Re}(\ell, \ell', x)]^T\}, x \in D_\ell \end{cases}$$

Conditions for accurate simulation [20, Assumption 2.5]. The data of the hybrid system $\mathcal{H} = (C, F, D, G)$, satisfies:

- A1. C and D are closed sets.
- A2. $F : \mathbb{R}^d \rightarrow \mathcal{P}(\mathbb{R}^d)$ is outer semicontinuous (o.s.c.)⁵ and locally bounded, and $F(\xi)$ is nonempty and convex for all $\xi \in C$.
- A3. $G : \mathbb{R}^d \rightarrow \mathcal{P}(\mathbb{R}^d)$ is o.s.c. and locally bounded, and $G(\xi)$ is nonempty for all $\xi \in D$.

Proposition 1 is an application of the following theorem. A ‘maximal’ trajectory is one which can not be extended. Recall that D is the time-limit of all trajectories we compute.

Theorem 4: [20, Thm. 3.4] Assume that $\mathcal{H}$ satisfies the above conditions. Let $K \subset \mathbb{R}^n$ be a compact set such that $\mathcal{H}$ is pre-complete from K (i.e. each maximal trajectory starting from K is either bounded or has infinite length). Then, for every $\epsilon > 0$ there exists $\delta^* > 0$ with the following property: for any $\delta \in (0, \delta^*]$ and any solution η_x with $x \in K + \delta B_1(0)$ there exists a solution η_{x_0} with $x_0 \in K$ such that the two trajectories have the same number of location transitions, and their continuous parts are ϵ -close at all times $t < D$ at which they’re in the same location.

⁵A set-valued map $F : \mathbb{R}^n \rightarrow \mathcal{P}(\mathbb{R}^n)$ is o.s.c. iff for all sequences $(\xi_i) \in \mathbb{R}^n$ converging to ξ and all sequences $(\omega_i) \in F(\xi_i)$ converging to ω , it holds that $\omega \in F(\xi)$.