

# Random Relaxation Abstractions for Bounded Reachability Analysis of Linear Hybrid Automata

Distributed Randomized Abstractions in Model Checking

Sumit Kumar Jha

School of Computer Science  
Carnegie Mellon University  
Email: jha@cs.cmu.edu

Susmit Jha

Electrical Engineering and Computer Science Department  
University of California Berkeley  
Email: jha@eecs.berkeley.edu

**Abstract**—The state of the art in the validation of linear hybrid automata has been restricted to systems with tens of variables because of the extremely high computational complexity of manipulating polyhedra in high dimensions. In this paper, we present a distributed algorithm that constructs low dimensional randomized over-approximate relaxation abstractions of linear hybrid automata and analyzes these low dimensional hybrid automata to perform *bounded model checking* of the original high dimensional linear hybrid automata. Our algorithm relies on the feasibility preserving nature of random linear relaxations and the *Johnson Lindenstrauss lemma* to show that random relaxations preserve the infeasibility of linear constraints with a nonzero probability.

## I. INTRODUCTION

The development of systems with high degree of correctness assurance needs the development of formal models for the analysis of these systems. Hybrid automata are a very well studied formalism for representing and analyzing dynamical systems with both discrete and continuous state variables. Such systems are widely prevalent and include cyber-physical systems, embedded software [1], mixed signal circuits [2], and quantitative models of biological systems [3]. Linear Hybrid Automata (LHA) are an important and algorithmically analyzable subclass of hybrid automata that can approximate hybrid automata with nonlinear continuous dynamics [4] asymptotically. LHA reachability analysis tools typically compute the sets of reachable states using polyhedra [5] explicitly, but the size of the polyhedral representations can be exponential in the number of continuous variables of the LHA. Thus, the verification of high-dimensional LHA is a hard problem. Although there has been considerable progress in the development of tools and algorithms for analyzing LHA [6]–[11], there is still a need for techniques to analyze high dimensional linear hybrid automata.

Our current technique extends earlier work on using *Iterative Relaxation Abstraction* (IRA) to prove the correctness of linear hybrid automata and the use of *Counterexample Guided Abstraction Refinement* (CEGAR) for analyzing embedded software. In our approach, several *randomized low dimensional over-approximate relaxation abstractions* of the original linear hybrid automata are constructed in a distributed manner.

Each random low dimensional relaxation abstraction is then analyzed by a traditional model checking algorithm [12], [13] to determine if there are any counterexamples in the low dimension abstraction. If the reduced *randomized low dimensional over-approximate relaxation abstraction* has no counterexamples, then the original concrete system is declared to satisfy the property. If a counterexample is found in the *randomized low dimensional over-approximate relaxation abstraction*, we use linear programming to check the feasibility of this counterexample in the original *concrete system*. If the counterexample is validated in the high dimension *concrete system*, the algorithm stops. Otherwise, we continue to generate more *randomized low dimensional over-approximate relaxation abstractions*.

The following are our new contributions to the verification of linear hybrid automata:

- We demonstrate the construction of *randomized low dimensional over-approximate relaxation abstractions* of linear hybrid automata that preserve the feasible paths of *bounded length* of the high dimensional linear hybrid automata.
- We show that the *randomized low dimensional over-approximate relaxation abstractions* of linear hybrid automata preserve the infeasibility of spurious counterexamples with nonzero probability. Our proof makes use of the notion of *irreducible infeasible subsets* and the *Johnson Lindenstrauss lemma*.
- We analyze the running time of our algorithm and show that the correctness guarantees of our algorithm improve with the construction of every *randomized low dimensional over-approximate relaxation abstractions*. Further, we argue that our algorithm is completely distributed and trivially parallelizable by design.

Our current approach extends and complements earlier work on model checking of linear hybrid automata [6]–[8], [11], [14] and combines it with the use of random linear relaxations that preserve the satisfiability of feasible linear constraints.

## II. RELATED WORK

The approach we suggest is different from the standard counterexample guided abstraction refinement (CEGAR) loop. In the CEGAR approach [15], a sequence of small overapproximate abstractions with small sets of variables is built unless a feasible counterexample is found or the abstraction is sufficient to prove the property. The subset of variables used to build the next abstraction corresponds to an unsatisfiable core returned by a SAT solver [16] or it may be found by heuristics in some cases [17]. These variables are then added to the set of variables used thus far to build the small abstraction and a new slightly larger abstraction is constructed. On each iteration, the abstraction therefore becomes more refined and larger in size. In the case of the verification of linear hybrid automata, bounded model checking has been experimentally observed to be doubly exponential in the dimension of the linear hybrid automata. Thus, even a moderate growth in the dimension of the abstraction makes it impossible to apply the CEGAR algorithm for the verification of linear hybrid automata.

The power of the CEGAR approach derives from its construction of a single sequence of more refined abstractions with a small number of variables for which model checking is tractable, while leveraging the power of convex decision procedures to deal with constraints involving many variables to test the validity of potential counterexamples in the original high-dimensional system. In a recent paper [11], a modified approach using linear programming has been suggested that builds upon the CEGAR algorithm to prove the correctness of a high dimensional linear hybrid automata with promising results. The use of a hierarchy of abstractions for analyzing lazy linear hybrid automata has also been studied recently [18].

While counterexample guided abstraction refinement algorithms were designed for discrete systems and the early adaptation of the idea to the continuous case [11], [14] has been relatively straightforward, these approaches use computationally expensive procedures like the *Fourier-Motzkin procedure* to construct the abstractions. Our current algorithm uses the power of random projections and does not need to invoke computationally expensive abstraction algorithms.

Our algorithm provides a framework for applying *distributed and randomized abstraction refinement* to the problem of verification of linear hybrid automata and a practical implementation of our approach should also use other complementary techniques including symbolic techniques [6], [7] and clever insights into the nature of the runs of a linear hybrid automata [8].

## III. DEFINITIONS

### A. Linear Constraints and Irreducible Infeasible Subsets

*Definition 1 (Linear Constraint):* A linear constraint of dimension  $m$  is a triple  $l = (c, \sim, b)$  where  $c = [c_1, \dots, c_m]^T \in \mathbb{R}^m$ ,  $\sim \in \{\geq, \leq\}$ , and  $b \in \mathbb{R}$ .

Given a column vector of variables  $X = [X_1, \dots, X_m]^T$ ,  $l_X : c^T X \sim b$  defines a linear constraint over  $X$ . For a given  $x \in \mathbb{R}^m$ ,  $l_X(x)$  denotes the boolean value of the expression

$l_X$  for the valuation  $X = x$ . Also, let  $2^A$  denotes the set of finite subsets of a set  $A$ . For  $P \in 2^A$ , the predicate  $P_X$  is defined by the conjunction of the constraints in  $P$  i.e.  $P_X \equiv \bigwedge_{l \in P} l_X$ . The predicate  $P_X$  corresponds to a (possibly open) polyhedron in  $\mathbb{R}^m$  denoted by  $\llbracket P \rrbracket$ .  $L^m$  denotes the set of all linear constraints of dimension  $m$ .

*Definition 2 (Satisfiability of Linear Constraint):* A set of linear constraints  $P \subset L^m$  is said to be *satisfiable* if and only if  $\llbracket P \rrbracket \neq \emptyset$ .

If  $P$  is not satisfiable, we are often interested in finding a small subset of constraints in  $P$  that is not satisfiable.

*Definition 3 (Irreducible Infeasible Subset [19]):* An *irreducible infeasible subset* (IIS) of a set of constraints  $P$  is a subset  $P' \subseteq P$  such that

- $P'$  is not satisfiable, and
- for any  $l \in P'$ ,  $(P' - \{l\})$  is satisfiable.

Although the problem of finding a minimum IIS is NP hard [20], several linear programming packages implement efficient heuristic procedures to compute IISs that are minimal and often minimum (e.g., LINDO [21], CPLEX [22], IBM OSL [23], MINOS (IIS) [24]).

### B. Linear Hybrid Automata

*Definition 4 (Linear Hybrid Automata [5], [11]):* A *linear hybrid automaton* (LHA) [13] is a tuple  $H = (G, n, \iota, \phi, \gamma, \rho)$ , such that

- $G = (Q, q_0, Q_{bad}, \Sigma, E)$  is the *location graph* of  $H$ , where
  - $Q$  is a finite set of *locations*;
  - $q_0 \in Q$  is the *initial* location;
  - $Q_{bad} \subset Q$  is the set of *bad* locations;
  - $\Sigma$  is a finite set of *labels*;
  - $E \subseteq Q \times \Sigma \times Q$  is the finite set of *transitions*;
- $n$  is the number of *continuous state variables*,
- $\iota : Q \rightarrow 2^{L^n}$  identifies the *invariant* for each location.  $L^n$  is the set of linear constraints in  $n$ -dimensions and  $2^{L^n}$  denotes the set of finite subsets of  $L^n$ .
- $\phi : Q \rightarrow 2^{L^n}$  identifies the *flow constraints* for each location.
- $\gamma : E \rightarrow 2^{L^n}$  identifies the *guard* for each transition.
- $\rho : E \rightarrow 2^{L^{2n}}$  identifies the *jump relation* for each transition. Again,  $L^{2n}$  is the set of linear constraints in  $2n$ -dimensions and  $2^{L^{2n}}$  denotes the set of finite subsets of  $L^{2n}$ .

We assume that the labels on the transitions out of any location are all distinct; so, a sequence of labels uniquely identifies a sequence of locations in the linear hybrid automata.

*Definition 5 (Path in Linear Hybrid Automata):* A path in the location graph is a finite sequence of the form

$$\pi = q_0, \sigma_1, q_1, \sigma_2, q_2, \dots, \sigma_l, q_l$$

such that for all  $k$  ( $0 \leq k \leq l - 1$ ),  $(q_k, \sigma_{k+1}, q_{k+1}) \in E$  (transitions).

*Definition 6 (Trace in Linear Hybrid Automata):* Given a path  $\pi$  of the linear hybrid automata

$$\pi = q_0, \sigma_1, q_1, \sigma_2, q_2, \dots, \sigma_l, q_l$$

the trace  $tr(\pi)$  corresponding to the path is the sequence of event labels in the location graph along the path i.e.

$$tr(\pi) = \sigma_1, \sigma_2, \dots, \sigma_l$$

*Definition 7 (Run of Linear Hybrid Automata [11], [25]):*

A run corresponding to a path  $\pi = q_0, \sigma_1, q_1, \sigma_2, q_2, \dots, \sigma_l, q_l$  for an LHA  $H$  is a finite sequence of the form

$$q_0, x^0, \sigma_1, q_1, x^1, \sigma_2, q_2, x^2, \dots, \sigma_l, q_l, x^l$$

where for all  $k = 0, 1, \dots, l$ ,  $t_s^k \leq t \leq t_f^k$ , and

- $x^k : [t_s^k, t_f^k] \rightarrow R^n$  denotes the continuous evolution of the continuous state variables;
- $x^k(t) \in \llbracket \iota(q_k) \rrbracket$  (location invariant polyhedra);
- $\dot{x}^k \in \llbracket \phi(q_k) \rrbracket$  (flow invariant polyhedra);

and for all  $k = 0, 1, \dots, l-1$

- $x^k(t_f^k) \in \llbracket \gamma(q_k, \sigma_{k+1}, q_{k+1}) \rrbracket$  (guard polyhedra);
- $(x^k(t_f^k), x^{k+1}(t_s^{k+1})) \in \llbracket \rho(q_k, \sigma_k, q_{k+1}) \rrbracket$  (jump relation polyhedra).

The above set of linear constraints are referred to as  $\mathcal{C}(H, \pi)$ .

*Definition 8 (Path Counterexample):* Given a linear hybrid automata  $H$ , a path

$$\pi = q_0, \sigma_1, q_1, \sigma_2, q_2, \dots, \sigma_l, q_l$$

for which  $q_l \in Q_{bad}$  is called a *path counterexample*.

*Definition 9 (Feasible and Spurious Counterexamples):* A path counterexample

$$\pi = q_0, \sigma_1, q_1, \sigma_2, q_2, \dots, \sigma_l, q_l$$

is said to be a *feasible counterexample* for  $H$  if there exists a run of the linear hybrid automata  $H$  of the form:

$$q_0, x^0, \sigma_1, q_1, x^1, \sigma_2, q_2, x^2, \dots, \sigma_l, q_l, x^l$$

Path counterexamples that are not feasible for  $H$  are called *spurious counterexamples*.

In an earlier paper, we demonstrated the fact that the existence of a *run* for a given path or trace of the linear hybrid automata can be formulated as the feasibility of a linear program [25].

*Lemma 1 (Counterexamples and Linear Programs [25]):*

A counterexample  $\pi$  is feasible in a linear hybrid automata  $H$  if and only if the set of constraints  $\mathcal{C}(H, \pi)$  (Definition 7) is feasible.

As we earlier noted, there is a one-to-one correspondence between paths and traces of a linear hybrid automata. Consequently, we will use paths and traces interchangeably. We denote by  $\mathcal{L}_{CE}(H)$  the language of all feasible counterexamples of the linear hybrid automata  $H$ . Following the terminology in the CEGAR literature, we refer to the given LHA  $H$  as the *concrete LHA* to distinguish it from its randomized relaxation

abstractions. We note that all relaxation abstractions have the same path counterexamples as the associated concrete LHA since all of these LHA have the same location graph.

### C. Randomized Relaxations

*Definition 10 (Randomizing Matrix):* An  $n \times d$  matrix  $M = \{M_{ij}\}$  is said to be a *randomizing matrix* if and only if each entry  $M_{ij}$  of the matrix is drawn *randomly* from a given probability distribution.

*Definition 11 (Low Distortion Randomizing Matrix):* Let  $X$  be an arbitrary set of  $|X|$  points in  $R^n$ , represented as an  $|X| \times n$  matrix  $A$ . Given  $\eta, \beta > 0$  and

$$d \geq \frac{4 + 2\beta}{\eta^2/2 - \eta^2/3} \log |X|$$

then the  $n \times d$  matrix  $M = \{M_{ij}\}$  is said to be a *low distortion randomizing matrix* for  $X$  iff

- $M_{ij} = +\sqrt{\frac{3}{n}}$  with probability  $1/6$ ;
- $M_{ij} = -\sqrt{\frac{3}{n}}$  with probability  $1/6$ ;
- $M_{ij} = 0$  with probability  $2/3$ .

*Definition 12 (Random Relaxation of Linear Constraints):* Given a set of linear constraints of dimension  $n$  i.e.  $P \subset L^n$  and a *randomizing matrix*  $M_{n \times d}$ , the set  $P \downarrow_M \equiv \{([c'_1, c'_2, \dots, c'_d], \sim, b) \mid ([c_1, c_2, \dots, c_n], \sim, b) \in P \text{ and } c'_i = \sum_k c_k M_{ki}\}$  is a set of linear constraints of dimension  $d$ . We call  $P \downarrow_M$  the *relaxation of  $P$  with respect to the randomizing matrix  $M$* .

*Definition 13 (Random Relaxation of LHA):* Given a linear hybrid automata  $H = (G, n, \iota, \phi, \gamma, \rho)$  and a *randomizing matrix*  $M_{n \times d}$ , the set of constraints  $H' = \{(\mathcal{C}(H, \pi)) \downarrow_M \mid \pi \text{ is a path in } H \text{ of length } k\}$  is said to be a *relaxation of  $H$  with respect to  $M$  unto depth  $k$* , denoted as  $H' = H \downarrow_M$ .

Note that the constraints need not be enumerated one path at a time and graph based approaches can be used to summarize the set of constraints as a boolean formula over linear inequalities. However, this definition suffices for the purpose of our proofs.

*Definition 14 (IIS Preserving Set):* Given an infeasible set of constraints  $P$  with an *irreducible infeasible subset*  $I = \{i_1, i_2, \dots, i_{|I|}\}$  of  $P$ , the set  $X = \{x_1, y_1, x_2, y_2, \dots, x_m, y_m\}$  formed by the vertices  $\{x_1, x_2, \dots, x_m\}$  of the polyhedra  $[[I \setminus \{i_1\}]]$  and the corresponding nearest points  $\{y_1, y_2, \dots, y_m\}$  in  $i_1$  is called the *IIS Preserving Set* for  $I$ .

*Definition 15 (IIS Preserving Randomizing Matrix):* Given an infeasible set of constraints  $P$  with an *irreducible infeasible subset*  $I$  of  $P$  and the *IIS Preserving Set*  $X$  for  $I$ , a *randomizing matrix*  $M$  is said to be an *IIS Preserving Randomizing Matrix* for  $I$  if and only if  $M$  preserves the pairwise distances among the points in  $X$ .

Intuitively, the points in  $X$  are a proof that the feasible subset of the IIS i.e.  $I \setminus \{i_1\}$  lies completely outside the region satisfying  $i_1$ . The *Johnson Lindenstrauss lemma* can be used to show that a *low distortion randomizing matrix* for  $X$  is an

IIS Preserving Randomizing Matrix for  $I$  with probability at least  $1 - |X|^{-\beta}$ .

#### IV. RESULTS

We first recall the *Johnson-Lindenstrauss lemma* which proves the existence of a projection from an  $n$ -dimensional space to a lower  $d$ -dimensional space that does not considerably distort the pairwise distance between any two points of a given set in the high dimension during the projection to the lower dimension.

*Lemma 2 (Johnson-Lindenstrauss [26]):* Given  $0 < \eta < 1$ , a set  $X$  of  $|X|$  points in  $\mathbb{R}^n$ , and a number  $d > n_0 = 4 \frac{\ln(|X|)}{\eta^2 - \eta^4}$ , there is a Lipschitz function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^d$  such that

$$(1 - \eta) \|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \eta) \|u - v\|^2$$

for all  $u, v \in X$ .

*Lemma 3 (Johnson-Lindenstrauss-Achlioptas [27]):* Let  $X$  be an arbitrary set of  $|X|$  points in  $\mathbb{R}^n$ , represented as an  $|X| \times n$  matrix  $A$ . Given  $\eta, \beta > 0$ ,  $d \geq \frac{4+2\beta}{\eta^2/2 - \eta^2/3} \log |X|$  and a *low distortion randomizing matrix*  $M$ , the function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^d$  mapping the  $i^{\text{th}}$  row of  $A$  to the  $i^{\text{th}}$  row of  $A \times M$  satisfies the following constraint with probability at least  $1 - |X|^{-\beta}$ , for all  $u, v \in X$

$$(1 - \eta) \|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \eta) \|u - v\|^2$$

*Theorem 1 (Random Relaxations of Feasible Constraints):*

If  $P \equiv \bigwedge_i l_i$  is a satisfiable set of linear constraints of dimension  $n$  and  $M_{n \times d}$  is a *randomizing matrix* and  $P' = P \downarrow_M$ , then the following holds:

$$[[P]] \neq \emptyset \implies [[P']] \neq \emptyset$$

*Proof:* Without loss of generality, consider any point  $x$  in the interior of the polyhedra  $[[P]]$ , it is sufficient to show that  $xM \in [[P']]$ .

Now,  $xM$  satisfies  $cx \sim b$  iff  $\sum_i c_i (xM)_i \sim b$  iff  $\sum_i c_i (\sum_j x_j M_{ij}) \sim b$  iff  $\sum_j (\sum_i c_i M_{ij}) x_j \sim b$ . Thus,  $x$  satisfies  $\sum_j (\sum_i c_i M_{ij}) x_j \sim b$  if and only if  $xM$  satisfies  $\sum_i c_i x_i \sim b$ .

$$\begin{aligned} x \in [[P]] &\implies x \in [[\bigwedge_i l_i]] \\ &\implies \bigwedge_i l_i x \\ &\implies \bigwedge_i l_i \downarrow_M xM \\ &\implies xM \in [[\bigwedge_i l_i \downarrow_M]] \\ &\implies xM \in [[P \downarrow_M]] \\ &\implies xM \in [[P']] \end{aligned}$$

*Lemma 4:* If  $I = \{i_1, i_2 \dots i_{|I|}\}$  is an *irreducible infeasible subset*,  $z$  is a point satisfying  $\neg i_1$ ,  $z'$  is a point satisfying  $I \setminus \{i_1\}$ , and  $M$  is an *IIS Preserving Randomizing Matrix* for  $I$ , then  $\|zM - z'M\| \neq 0$ .

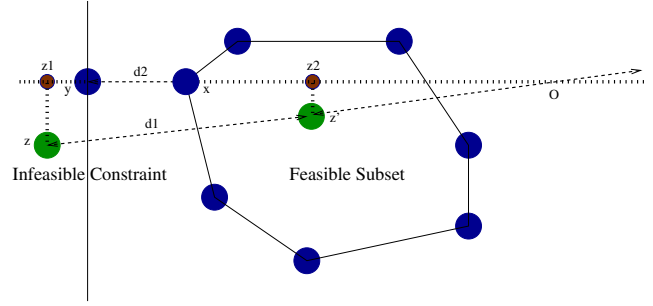
*Proof:* Consider a vertex  $x$  of  $I \setminus \{i_1\}$  that is at least as close to  $i_1$  as  $z'$ . Also, let  $y$  be the nearest point on  $i_1$  from  $x$ . Since,  $I$  is an IIS,  $\|x - y\| \neq 0$ . Let  $z_2$  be the projection of  $z'$  and  $z_1$  be that of  $z$  on the line through  $x$  and  $y$ . Let  $d_2$  be the distance between  $z_1$  and  $z_2$ , and  $d_1$  be the distance between  $z$  and  $z'$ .

Then,  $\|z_2 - x\| \geq 0$  (Since  $z'$  lies inside the polyhedra and  $x$  is as close to  $i_1$  as  $z'$ .)

Also,  $\|z_1 - y\| \geq 0$  (Since,  $y$  lies on  $i_1$  and  $z$  lies on or inside  $i_1$ .)

Now, from the plane geometry,  $d_1 \geq d_2$ . (Consider the right triangle formed by  $Oz'z_2$  and  $Ozz_1$ , use similarity of triangles and the fact that the hypotenuse is longer than any of the sides.)

$$\text{Thus, } \|z' - z\| = d_1 \geq d_2 \geq \|x - y\|$$

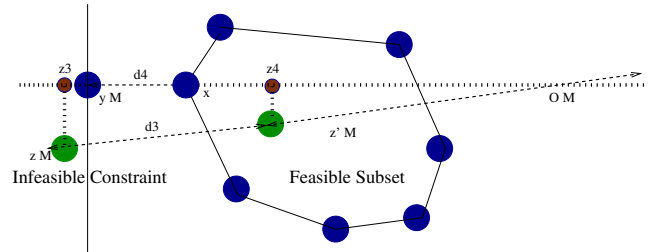


Consider the points  $y, x, O$  and  $z, z', O$  after the projection  $M$ . As the projection  $M$  is linear, the points  $yM, xM$  and  $OM$  lie on a straight line, and so do  $zM, z'M$  and  $OM$ . Then, the following holds due to the same argument as above:

$$\|z'M - zM\| \geq \|xM - yM\|.$$

Also, because  $M$  is an *IIS Preserving Randomizing Matrix*,  $\|xM - yM\| \geq \|x - y\|/\eta$ .

$$\text{Thus, } \|z'M - zM\| \geq \|x - y\|/\eta \neq 0.$$



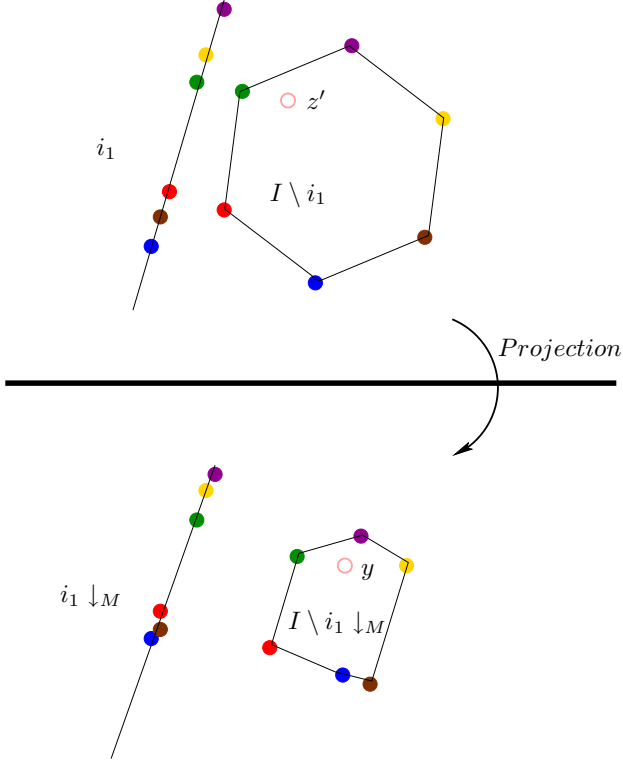
*Lemma 5:* If  $I = \{i_1, i_2 \dots i_{|I|}\}$  is an *irreducible infeasible subset* of an unsatisfiable set of linear constraints  $P$ ,  $M$  is an *IIS Preserving Randomizing Matrix* for  $I$ ,  $y$  is a point in the interior of the feasible projection  $(I \setminus \{i_1\}) \downarrow_M$  i.e.  $y \in [(I \setminus \{i_1\}) \downarrow_M]$  and  $z$  be a point such that  $y = zM$ , then  $z$  satisfies  $\neg i_1$ .

*Proof:*

First we show that there exists a  $z' \in [(I \setminus \{i_1\})]$  such that  $y = z'M$ .

Now,  $xM$  satisfies  $cx \sim b$  iff  $\sum_i c_i (xM)_i \sim b$  iff  $\sum_i c_i (\sum_j x_j M_{ij}) \sim b$  iff  $\sum_j (\sum_i c_i M_{ij}) x_j \sim b$ . Thus,  $x$  satisfies  $\sum_j (\sum_i c_i M_{ij}) x_j \sim b$  if and only if  $xM$  satisfies  $\sum_i c_i x_i \sim b$ .

$$\begin{aligned}
& y \in [(I \setminus \{i_1\}) \downarrow_M] \\
& \implies y \in [[\bigwedge_{i>1} l_i \downarrow_M]] \\
& \implies \bigwedge_{i>1} l_i \downarrow_M (y) \\
& \implies \bigwedge_{i>1} l_i z' \text{ (where } y = z'M \text{ from linearity} \\
& \text{of the transformation)} \\
& \implies z' \in [[\bigwedge_{i>1} l_i]] \\
& \implies z' \in [(I \setminus \{i_1\})]
\end{aligned}$$



Further, suppose (to the contrary of the lemma) that  $z$  satisfies  $i_1$  and its projection  $y (= zM)$  lies in  $[(I \setminus \{i_1\}) \downarrow_M]$ . Since  $z \neq z'$ ,  $\|z - z'\| \neq 0$ . But, for an *IIS Preserving Randomizing Matrix*  $M$ ,  $\|zM - z'M\| \neq 0$  (from Lemma 4) i.e. if  $z$  satisfies  $i_1$ ,  $zM \neq z'M$  i.e.  $zM \neq y$ . Hence, by contradiction,  $z$  does not satisfy  $i_1$  i.e.  $z$  satisfies  $\neg i_1$ . ■

**Theorem 2 (Relaxations of Infeasible Constraints):** If  $P \equiv \bigwedge_i l_i$  is an unsatisfiable set of linear constraints of dimension  $n$  with an *irreducible infeasible subset* (IIS)  $I = \{i_1, i_2, \dots, i_{|I|}\}$ ,  $X$  is the *IIS Preserving Set* of the IIS  $I$ , and  $M_{n \times d}$  is an *IIS Preserving Randomizing Matrix* for  $I$ , then the following holds:

$$[[P]] = \emptyset \implies [[P \downarrow_M]] = \emptyset$$

*Proof:* We know that  $[[P]] = \emptyset$  with the IIS  $I$  and the *IIS Preserving Set*  $X$ . Intuitively, the points in  $X$  are a proof that the feasible subset of the IIS i.e.  $I \setminus \{i_1\}$  lies completely outside the region satisfying  $i_1$ .

Since,  $I \setminus \{i_1\}$  is feasible,  $(I \setminus \{i_1\}) \downarrow_M$  is feasible too (from Theorem 1). Consider a point  $y$  in the interior of the feasible

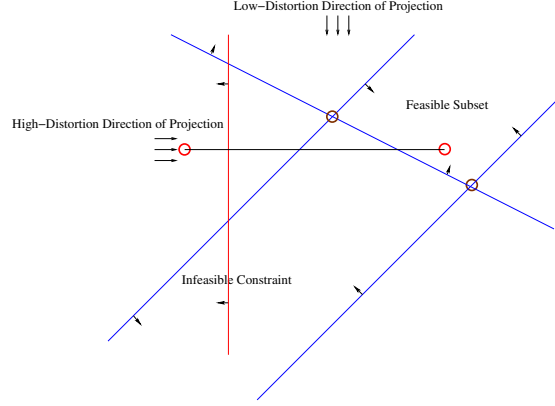


Fig. 1: Examples of Low and High Distortion Projections of an IIS

projection  $(I \setminus \{i_1\}) \downarrow_M$  i.e.  $y \in [(I \setminus \{i_1\}) \downarrow_M]$ . If  $M$  is an *IIS Preserving Randomizing Matrix* and  $z$  is a point such that  $y = zM$ ,  $z$  satisfies  $\neg i_1$  (Lemma 5) and hence,  $y$  satisfies  $\neg i_1 \downarrow_M$  (by Theorem 1). Thus, all points in  $[(I \setminus \{i_1\}) \downarrow_M]$  satisfies  $\neg i_1 \downarrow_M$ . Hence,  $I \downarrow_M$  is infeasible and so is  $P \downarrow_M$ . ■

**Corollary 1 (Relaxations of Infeasible Constraints):** If  $P \equiv \bigwedge_i l_i$  is an unsatisfiable set of linear constraints of dimension  $n$  with an *irreducible infeasible subset* (IIS)  $I = \{i_1, i_2, \dots, i_{|I|}\}$ ,  $X$  is the *IIS Preserving Set* of the IIS  $I$ , and  $M_{n \times d}$  is an *low distortion randomizing matrix* for  $I$ , then the following holds with probability at least  $1 - |X|^{-\beta}$ :

$$[[P]] = \emptyset \implies [[P \downarrow_M]] = \emptyset$$

**Theorem 3 (Counterexamples in Random Relaxations):** If  $M_{m \times n} = \{M_{ij}\}$  be a randomizing matrix and  $\pi$  is a feasible counterexample in  $H$  i.e.  $\pi \in \mathcal{L}_{CE}(H)$ , then  $\mathcal{C}(H, \pi) \downarrow_M$  is feasible.

*Proof:*

Consider a path  $\pi \in \mathcal{L}_{CE}(H)$  such that

$$\pi \equiv q_0, \sigma_1, q_1, \sigma_2, q_2, \dots, \sigma_l, q_l$$

We know that there is a conjunction of linear constraints  $\mathcal{C}(H, \pi)$  such that  $\pi$  is a feasible counterexample *only if*  $\mathcal{C}(H, \pi)$  is feasible. Since  $\mathcal{C}(H', \pi)$  is feasible,  $\mathcal{C}(H, \pi) \downarrow_M$  is feasible too (Theorem 1). ■

Consequently, all concrete counterexamples are also feasible counterexamples for any random relaxation abstraction.

**Theorem 4 (Infeasible Counterexamples in Relaxations):** If  $M_{m \times n} = \{M_{ij}\}$  be a *low distortion randomizing matrix* preserving the set  $X$ , and  $\pi$  is not a feasible counterexample in  $H$  i.e.  $\pi \notin \mathcal{L}_{CE}(H)$ , then  $\mathcal{C}(H, \pi) \downarrow_M$  is infeasible with probability at least  $1 - |X|^{-\beta}$ .

*Proof:*

Consider a path  $\pi \in \mathcal{L}_{CE}(H)$  such that

$$\pi \equiv q_0, \sigma_1, q_1, \sigma_2, q_2, \dots, \sigma_l, q_l$$

We know that there is a conjunction of linear constraints  $\mathcal{C}(H, \pi)$  such that  $\pi$  is an infeasible counterexample *only if*  $\mathcal{C}(H, \pi)$  is infeasible. Since  $\mathcal{C}(H, \pi)$  is infeasible,  $\mathcal{C}(H, \pi) \downarrow_M$  is infeasible too with probability at least  $1 - |X|^{-\beta}$  (Corollary 1).

## V. DISTRIBUTED ALGORITHM

Our distributed algorithm consists of the following steps:

- Set  $Iteration = 0$ ;
- Let each of the  $D$  distributed nodes *independently* compute a randomized relaxation  $H_i$  of the constraints denoting the *bounded reachability* of the  $n$ -dimensional linear hybrid automata  $H$  into a constraint denoting the reachability of the linear hybrid automata of reduced dimension  $d$ .
- Each randomized relaxation  $H_i$  is a low-distortion relaxation refuting all spurious counterexamples with probability  $1 - |X|^{-\beta}$ , where  $X$  is twice the sum of the sizes of the vertices of all the IISs and  $\beta$  is a constant in the Johnson-Lindenstrauss lemma.
- For each random relaxation, compute the counterexample language  $\mathcal{L}(H_i)$  of the low dimensional LHA using standard reachability.
- Compute the global *Counterexample Abstraction*  $CE = CE \cap \mathcal{L}(H_0) \cdots \cap \mathcal{L}(H_D)$ .
- If  $CE$  is empty, report that the system is safe and stop.
- If  $CE$  is not empty, pick a counterexample  $ce$  and test if it is feasible in the *concrete system* using simulation.
- If  $CE$  is feasible in the *concrete system*, stop and report the counterexample.
- Otherwise, all spurious counterexamples have been shown to be infeasible with probability at least  $1 - |X|^{-\beta}$  by each node;
- Set  $Iteration = Iteration + 1$ ; continue building more abstractions from Step 3.

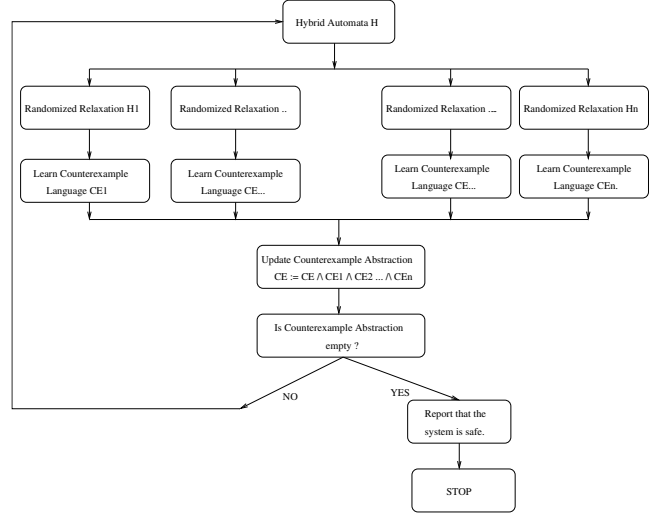
*Theorem 5 (Bound on the Probability of One-Sided Error):* If the algorithm was run on  $D$  nodes for  $L$  iterations and no counterexample could be found, the system is indeed correct with probability at least  $1 - (|X|^{-\beta DL})$ .

*Proof:* If a counterexample was present, each node would have found it with probability at least  $1 - (|X|^{-\beta})$  (by refuting all the spurious counterexamples). As all the nodes and the iterations are independent, the probability that none of them would have found the counterexample is at most  $(|X|^{-\beta DL})$ . Thus, the probability that one of them would have found the counterexample is at least  $1 - (|X|^{-\beta DL})$ .

## VI. EXPERIMENTS

While our algorithm has sound theoretical guarantees, we present some examples to illustrate the benefit from using *low distortion random relaxations*.

We implemented our distributed randomization based relaxation abstraction approach using the state of the art linear programming solver available in MATLAB. The LP solver



No. of Constraints	Dimension	Reduced Dimension	Time (Original)	Time (Abstraction)	No. of Abstractions	Confidence
700	150 (Feasible)	60	6.7390	4.9818	17	99.99 %
1500	150 (Feasible)	60	18.6570	7.0089	17	99.99 %
1500	200 (Feasible)	60	23.6840	8.3721	17	99.99 %
1500	500 (Feasible)	60	62.3790	8.2738	17	99.99 %
10000	500 (Feasible)	100	Out of Memory	46.8847	3	99.99 %
10000	200 (Infeasible)	60	181.9420	110.5290	1	100%
5000	300 (Infeasible)	60	120.2630	48.3300	1	100%
10000	500 (Infeasible)	60	Out of Memory	132.3509	1	100%

TABLE I: Experimental Results: Time reported is in seconds.

could not solve many large problems with 500 variables and 10000 constraints directly but was able to solve these problems within a few minutes with the help of our distributed abstraction framework. Even on problems where the LP solver succeeded, the distributed randomization based relaxation abstraction outperformed the LP solver working on its own.

## VII. CONCLUSION

The paper presents a distributed randomized algorithm to efficiently verify linear hybrid automata by constructing low dimensional random relaxations of high dimensional linear hybrid automata. The linearity of the random relaxations preserves the feasibility of real counterexamples in all random abstractions while *Johnson Lindenstrauss lemma* assures that infeasible counterexamples remain infeasible in random abstractions with high probability. Our algorithm learns from

the random abstractions and performs an efficient bounded reachability verification of linear hybrid automata. Our results also provide a characterization of the hardness of linear hybrid automata reachability problems in terms of the irreducible infeasible subsets of the spurious counterexamples and hence, have theoretical value by themselves.

#### ACKNOWLEDGMENT

The second author acknowledges the support of the Berkeley Fellowship for Graduate Studies from UC Berkeley.

#### REFERENCES

- [1] Array, "Automotive engine control and hybrid systems: Challenges and opportunities," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 888–912, July 2000. [Online]. Available: <http://www.gigascale.org/pubs/66.html>
- [2] S. Little, N. Seegmiller, D. Walter, C. Myers, and T. Yoneda, "Verification of analog/mixed-signal circuits using labeled hybrid petri nets," in *ICCAD '06: Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*. New York, NY, USA: ACM Press, 2006, pp. 275–282.
- [3] P. Lincoln and A. Tiwari, "Symbolic systems biology: Hybrid modeling and analysis of biological networks," in *Hybrid Systems: Computation and Control HSCC*, ser. LNCS, R. Alur and G. Pappas, Eds., vol. 2993. Springer, Mar. 2004, pp. 660–672.
- [4] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, no. 1, pp. 3–34, 1995. [Online]. Available: [citeseer.ist.psu.edu/alur95algorithmic.html](http://citeseer.ist.psu.edu/alur95algorithmic.html)
- [5] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "HyTech: A model checker for hybrid systems," *International Journal on Software Tools for Technology Transfer*, vol. 1, no. 1–2, pp. 110–122, 1997. [Online]. Available: [citeseer.ist.psu.edu/henzinger97hytech.html](http://citeseer.ist.psu.edu/henzinger97hytech.html)
- [6] F. Wang, "Efficient verification of timed automata with bdd-like data structures," *Int. J. Softw. Tools Technol. Transf.*, vol. 6, no. 1, pp. 77–97, 2004.
- [7] F. Wang and H.-C. Yen, "Reachability solution characterization of parametric real-time systems," *Theor. Comput. Sci.*, vol. 328, no. 1–2, pp. 187–201, 2004.
- [8] X. Li, J. Zhao, T. Zheng, Y. L. 0005, and G. Zheng, "Duration-constrained regular expressions," *Formal Asp. Comput.*, vol. 16, no. 2, pp. 155–163, 2004.
- [9] R. Alur, T. Henzinger, and H. Wong-Toi, "Symbolic analysis of hybrid systems," in *Proc. 37-th IEEE Conference on Decision and Control*, 1997. [Online]. Available: [citeseer.ist.psu.edu/alur97symbolic.html](http://citeseer.ist.psu.edu/alur97symbolic.html)
- [10] G. Frehse, "PHAVer: Algorithmic Verification of Hybrid Systems Past HyTech," in *HSCC*, ser. Lecture Notes in Computer Science, M. Morari and L. Thiele, Eds., vol. 3414. Springer, 2005, pp. 258–273.
- [11] S. K. Jha, B. H. Krogh, J. E. Weimer, and E. M. Clarke, "Reachability for linear hybrid automata using iterative relaxation abstraction," in *HSCC*, ser. Lecture Notes in Computer Science, A. Bemporad, A. Bicchi, and G. C. Buttazzo, Eds., vol. 4416. Springer, 2007, pp. 287–300.
- [12] J. E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. Cambridge, MA, USA: MIT Press, 1999.
- [13] P.-H. Ho, "Automatic Analysis of Hybrid Systems, Ph.D. thesis, technical report CSD-TR95-1536, Cornell University, August 1995, 188 pages," 1995. [Online]. Available: <http://mtc.epfl.ch/~tah/Students/ho.pdf>
- [14] S. K. Jha, "d-ira: Distributed iterative relaxation abstraction for reachability analysis of linear hybrid automata," in *HSCC*, ser. Lecture Notes in Computer Science, A. Bemporad, A. Bicchi, and G. C. Buttazzo, Eds., vol. 4416. Springer, 2008, pp. 300–308.
- [15] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, "Counterexample-guided abstraction refinement," in *CAV '00: Proceedings of the 12th International Conference on Computer Aided Verification*. London, UK: Springer-Verlag, 2000, pp. 154–169. [Online]. Available: <http://portal.acm.org/citation.cfm?id=734089>
- [16] L. Zhang and S. Malik, "Validating SAT Solvers Using an Independent Resolution-Based Checker: Practical Implementations and Other Applications," in *DATE*. IEEE Computer Society, 2003, pp. 10 880–10 885.
- [17] S. Chaki, E. Clarke, A. Groce, and O. Strichman, "Predicate abstraction with minimum predicates," in *Proceedings of 12th Advanced Research Working Conference on Correct Hardware Design and Verification Methods (CHARME)*, 2003. [Online]. Available: [citeseer.ist.psu.edu/chaki03predicate.html](http://citeseer.ist.psu.edu/chaki03predicate.html)
- [18] S. Jha, B. A. Brady, and S. A. Seshia, "Symbolic reachability analysis of lazy linear hybrid automata," in *FORMATS*, ser. Lecture Notes in Computer Science, J.-F. Raskin and P. S. Thiagarajan, Eds., vol. 4763. Springer, 2007, pp. 241–256.
- [19] J. Chinneck and E. Dravnieks, "Locating minimal infeasible constraint sets in linear programs," *ORSA Journal on Computing*, vol. 3, pp. 157–168, 1991.
- [20] J. K. Sankaran, "A note on resolving infeasibility in linear programs by constraint relaxation," *Operations Research Letters*, vol. 13, p. 1920, 1993.
- [21] L. Systems Inc., "<http://www.lindo.com/products/api/dllm.html>," 2007.
- [22] ILOG, "<http://www.ilog.com/products/cplex/product/simplex.cfm>," 2007.
- [23] M. S. Hung, W. O. Rom, and A. D. Waren, "Optimization with IBM OSL and Handbook for IBM OSL," 1993.
- [24] J. W. Chinneck, "MINOS(IIS): Infeasibility analysis using MINOS," *Comput. Oper. Res.*, vol. 21, no. 1, pp. 1–9, 1994.
- [25] X. Li, S. K. Jha, and L. Bu, "Towards an Efficient Path-Oriented Tool for Bounded Reachability analysis of Linear Hybrid Systems using Linear Programming," 2006. [Online]. Available: [http://cs.nyu.edu.cn/people/lixuandong/BMC\\_complete.pdf](http://cs.nyu.edu.cn/people/lixuandong/BMC_complete.pdf)
- [26] S. Dasgupta and A. Gupta, "An elementary proof of the johnson-lindenstrauss lemma," Berkeley, CA, Tech. Rep. TR-99-006, 1999. [Online]. Available: <http://citeseer.ist.psu.edu/dasgupta99elementary.html>
- [27] D. Achlioptas, "Database-friendly random projections," in *PODS '01: Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. New York, NY, USA: ACM, 2001, pp. 274–281. [Online]. Available: <http://portal.acm.org/citation.cfm?id=375608>
- [28] A. Bemporad, A. Bicchi, and G. C. Buttazzo, Eds., *Hybrid Systems: Computation and Control, 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007, Proceedings*, ser. Lecture Notes in Computer Science, vol. 4416. Springer, 2007.