Reachset Conformance Testing of Hybrid Automata

Hendrik Roehm, Jens Oehlerking, Matthias Woehrle
Robert Bosch GmbH, Corporate Research
Renningen, Germany
{firstname.lastname}@de.bosch.com

Matthias Althoff
Department of Informatics
Technische Universität München, Germany
althoff@in.tum.de

ABSTRACT

Industrial-sized hybrid systems are typically not amenable to formal verification techniques. For this reason, a common approach is to formally verify abstractions of (parts of) the original system. However, we need to show that this abstraction conforms to the actual system implementation including its physical dynamics. In particular, verified properties of the abstract system need to transfer to the implementation. To this end, we introduce a formal conformance relation, called reachset conformance, which guarantees transference of safety properties, while being a weaker relation than the existing trace inclusion conformance. Based on this formal relation, we present a conformance testing method which allows us to tune the trade-off between accuracy and computational load. Additionally, we present a test selection algorithm that uses a coverage measure to reduce the number of test cases for conformance testing. We experimentally show the benefits of our novel techniques based on an example from autonomous driving.

CCS Concepts

•Computing methodologies → Model verification and validation; •Software and its engineering → Software verification; Software safety; Software verification and validation; *Dynamic analysis;* **•Computer systems organization →** *Embedded systems;*

Keywords

Conformance; Testing; Reachability Analysis; Test Selection; Hybrid Automata

1. INTRODUCTION

Embedded software controls the evolution of the physical behaviour of systems through a perception-action loop. Typically, this software comes with safety-critical properties that should be verified. Since the software strongly interacts with the physical dynamics, the composed system has to be taken into account for typical verification tasks. Hybrid automata are a suitable modeling formalism for these systems that can be directly used to formally verify embedded software. However, formal verification is computationally expensive and becomes infeasible for larger models of embedded systems. For this reason, a common approach is to use abstractions that are amenable to formal verification [4, 9]. However, when properties are verified on an abstract model, we have to check that they also transfer to the real system, which is often ignored or done in a non-formal way. A formal conformance relation between the systems enables to transfer properties from the abstract model to the real system. Given a class of relevant properties, the conformance relation should be as permissive as possible, yet as strong as necessary to transfer these properties between systems. A major problem is that in practice the conformance between systems cannot be formally shown, because real systems and complex simulation models are typically not amenable to formal techniques. In contrast, conformance testing is possible, which means searching for counter-examples falsifying the considered relation. This is an important condition for the applicability of formal methods for industrial-sized problems, because it substantiates the confidence in the abstract model and the properties verified thereon. For (formal) conformance testing there are three main tasks: (*i*) formally defining the *conformance relation* and proving the *transference* of properties, (*ii*) establishing a sound *conformance check*, such that only true counter-examples are identified, (*iii*) selecting *test inputs*, which produce different be-

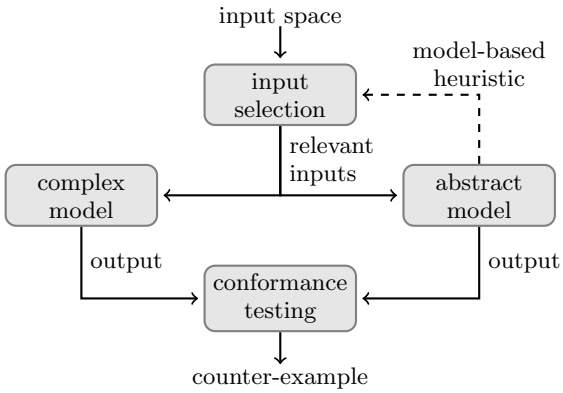


Figure 1: Overall structure of the proposed method.

HSCC'16, April 12 - 14, 2016, Vienna, Austria

ISBN 978-1-4503-3955-1/16/04…$15.00

DOI: http://dx.doi.org/10.1145/2883817.2883828

haviours, because only a limited, finite number of tests can be performed. In this work all three tasks are addressed.

It is essential that conformance testing is as formal as possible, e. g. to have a sound understanding which properties transfer with the given relation. The question in this paper is, which conformance relation should be used for safety properties and how conformance testing of this relation can be done.

Existing notions of conformance mainly determine if the traces of one system are contained in the set of traces of another, see Sec. 7. This is usually not an easy task and leads to very bloated and incomprehensible abstract models. Reachability analysis has been used for conformance testing, but the conformance relation has not been formally defined [4].

The contribution of this paper is a formal framework for conformance testing of hybrid automata considering safety properties, as shown in Fig. 1. Given a complex and an abstract model together with an input space, our method efficiently searches for counter-examples falsifying the reachset conformance relation. This is done by the following steps: (*i*) We introduce the formal definition of a conformance relation, called reachset conformance. The relation guarantees the transference of safety properties and is a weaker relation than the already existing trace conformance (cf. Sec. 3). (*ii*) We formalize the conformance testing approach of Althoff and Dolan [4] and extend it by using tighter overapproximations for inclusion checking and prove the soundness of the presented method. Therefore the trade-off between accuracy and computational load can be freely adapted and errors of simulations and measurements can be considered. (*iii*) We present a model-based input selection algorithm based on a reachset coverage measure. It can be used to reduce the number of tests for a given set of test cases. One benefit of the framework is the possibility to use measurements of a real system directly for falsification of the reachset conformance relation. Finally, we experimentally show that we are able to falsify more conformance relations between systems than the previous work by Althoff and Dolan [4].

In Sec. 2 we give basic definitions, such as hybrid automata, traces, and reachable sets. In Sec. 3 we introduce the formal definition of reachset conformance. We prove the transference of safety properties and the weakness compared to trace conformance. A method for reachset conformance testing for a given input is presented in Sec. 4. For the selection of relevant inputs, an algorithm is introduced in Sec. 5. The results of an autonomous driving example are shown in Sec. 6. Finally, we review the related work in Sec. 7 and give a conclusion in Sec. 8.

2. MODEL AND DEFINITIONS

We model hybrid systems as hybrid automata with inputs and outputs. Let $\|g\|_2$ be the Euclidean vector norm of a vector g and g^T be the transpose of a vector. We use $u(.)$ as a notation for an input trajectory and $u(t)$ as an input at time t.

Our definition of a *hybrid automaton* is a finite automaton whose discrete states are annotated with differential inclusions that define the evolution of the continuous states. Due to non-deterministic modeling, we use differential inclusions for the continuous flow resulting in infinitely many solutions for a given initial state. The initially possible states are given by the initial set and according to the continuous evolution and the input, the system can switch its discrete state. Here, we consider hybrid automata that take continuous input functions $u(.) : \mathbb{R}^+ \to \mathbb{R}^d$ from a set $U(.)$ of input functions to influence the evolution. For a more precise definition of hybrid automata, defining invariant sets, guard conditions, and reset maps, we refer to the work of Mitchell [18]. For simplicity, we assume that all hybrid automata are non-zeno and non-blocking and for every input there exists at least one solution.

A (state) solution x of the hybrid automaton S under a given input $u(.) \in U(.)$ is a trajectory that has the form

$$x = (q_0, x_0(.))(q_1, x_1(.)) \dots$$

where q_i are discrete states and $x_i : [t_i, t_{i+1}] \to \mathbb{R}^n$ is the continuous evolution between t_i and t_{i+1} with $t_0 = 0$ and $t_{i+1} \geq t_i$.

For one solution x, the output trace that is the mapping of the state solution onto the observable output space, is defined as $\tau : \mathbb{R}^+ \to \mathbb{R}^m$, where

$$\forall i \; \forall t \in [t_i, t_{i+1}) : \quad \tau(t) = O(q_i, x_i(t))$$

holds, where O is the output mapping. The set of all output traces under an input $u(.)$ is denoted by $\mathit{Traces}(S, u(.))$. If $\mathit{Traces}(S, u(.))$ has one element only for every $u(.)$, the system S is called deterministic. For a finite subset of time instances $T \subset \mathbb{R}^+$, the sampled trace of τ is the restriction to the preimage T

$$\tau_T : T \to \mathbb{R}^m, t \mapsto \tau(t).$$

For one point in time t, the reachable set of S at time t is defined as

$$\mathit{Reach}_t(S, u(.)) = \{\tau(t) \mid \tau \in \mathit{Traces}(S, u(.))\}$$

for a given input trajectory $u(.)$.

The elements of *Traces* are functions over time, whereas the set Reach_t consists of output states for one point in time t. Note that we define both in the output space, but not in the state space as done in other works (cf. [4]). We also consider a set of initial states but do not annotate this with a subscript. In the following, when we talk about systems, we assume they are modelled as hybrid automata.

3. REACHSET CONFORMANCE

Throughout the paper we use two systems S_r and S_a. The system S_r represents a real system or a complex simulation model that is not amenable to formal verification techniques. However, we can obtain measurements of executions or simulation runs for a given input. The system S_a is an abstract model that is simple enough to be used for formal verification. The main question here is if S_r conforms to S_a and which properties transfer.

First, we discuss the existing *trace conformance* relation used in [7]. Although it is a very strong relation that enables the transference of all properties which are $\forall$-quantified over the traces, it is also difficult to generate an abstract model S_a where it holds. If the focus is on the transference of safety properties, such as collision-free trajectories for autonomous vehicles, such a strong relation is not needed. Therefore, we define the weaker *reachset conformance* relation that is able to transfer such properties. This enables us to transfer safety properties between systems where the trace conformance does not hold.

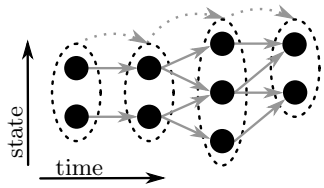


Figure 2: The reachable sets do not maintain the individual state transitions. Solid: States and transitions, dotted: Reachable sets and their transitions.

3.1 Trace conformance

In this subsection, we discuss the conformance relation used for instance by Dang [7] (cf. Sec. 7).

DEFINITION 1 (TRACE CONFORMANCE). *Let S_r and S_a be two systems with the same input set and output space, then S_r is trace conformant to S_a, denoted by $S_r \preceq_{Tr} S_a$, iff*

$$Traces(S_r, u(.)) \subseteq Traces(S_a, u(.))$$

holds for all $u(.) \in U(.)$.

The trace conformance reflects the conventional notion of conformance of discrete automata where traces of one system also have to be traces of the other. When the trace conformance does hold, all properties with an $\forall$-quantifier over traces, such as Metric Temporal Logic formulas, transfer (cf. [1]). However, considering safety properties only, for the trace conformance check of nondeterministic hybrid systems, we have to deal with two problems: (1) We have to check a relation that transfers more properties than we are interested in. Therefore we can relate less systems without any benefit. (2) For conformance testing we have to sample not only the input space but also the nondeterminism of the system leading to more traces needed for a test coverage. In the following subsection we define the reachset conformance relation to overcome the mentioned problems.

3.2 Reachset conformance

We now introduce the formal definition of a reachset conformance relation which is able to preserve safety properties, such as non-intersection with unsafe states. It is weaker than trace conformance and can be checked by applying the whole range of methods from reachability analysis.

Inspired by Althoff and Dolan [4], we formally define a new notion of conformance that focuses not on the set of traces, but on the set of reachable states.

DEFINITION 2 (REACHSET CONFORMANCE). *Let S_r and S_a be two systems with the same input set and output space, then S_r is reachset conformant to S_a, denoted by $S_r \preceq_R S_a$, iff*

$$Reach_t(S_r, u(.)) \subseteq Reach_t(S_a, u(.)) \tag{1}$$

holds for all $u(.) \in U(.)$ and $t \geq 0$.

The proposed reachset conformance allows the transference of safety properties from S_a to S_r:

PROPOSITION 1. *Let two systems S_r and S_a be given with $S_r \preceq_R S_a$. For any input trajectory $u(.)$ and any unsafe set B_t the following transference holds for every t:*

$$Reach_t(S_a, u(.)) \cap B_t = \emptyset \Rightarrow Reach_t(S_r, u(.)) \cap B_t = \emptyset.$$

Since the relation considers only reachsets, we do not have to maintain the individual dependences of each reachable state for one time instance to another as depicted in Fig. 2. Since trace conformance considers the entire signals, it is a stronger relation.

PROPOSITION 2. *Let S_r and S_a be two systems with the same input set and output space, then*

$$S_r \preceq_{Tr} S_a \quad \Rightarrow \quad S_r \preceq_R S_a \tag{2}$$

holds. The converse holds if the system S_a is deterministic.

PROOF. Let $u(.)$ be an input trajectory, t a point in time, and $y \in Reach_t(S_r, u(.))$ and $S_r \preceq_{Tr} S_a$. Then, there is a $\tau \in Traces(S_r, u(.))$ with $\tau(t) = y$. From $S_r \preceq_{Tr} S_a$ it follows, that τ is also a trace of S_a and $y \in Reach_t(S_a, u(.))$. The proposition follows, because the aforementioned implication holds for all y, t, and $u(.)$. When the system S_a is deterministic, there is only one trace in $Traces(S_a, u(.))$ and the reachable sets consist of only one state. Hence S_r has the same trace and is also deterministic. $\square$

The main difference between trace and reachset conformance consists in the handling of nondeterminism. In the following, we present an example to give a better understanding of the conformance notions and to show that the reverse implication of Eq. (2) does not hold in general.

EXAMPLE 1. *For the sake of simplicity, we pick two continuous systems without inputs. Let S_r be a 2-dim. system with $F((x_1, x_2)^T) = (x_2, -x_1)^T$, output map $O((x_1, x_2)^T) = x_1$, and initial set*

$$A = \{(x_1, x_2)^T \mid x_1^2 + x_2^2 = 0.5\}.$$

Then the set of traces is

$$Traces(S_r) = \{0.5 \sin(t + c) \mid c \in [-\pi, \pi)\}$$

and the reachable set is the time-invariant set

$$Reach_t(S_r) = [-0.5, 0.5] \quad \forall t \geq 0.$$

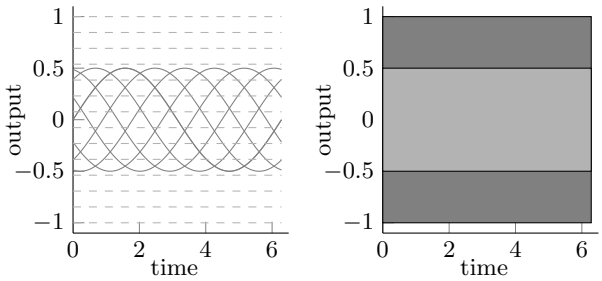
Let S_a be a 1-dim. abstract system with $F(x) = 0$, initial set $A = [-1, 1]$, and output map $O(x) = x$. Then the set of traces is

$$Traces(S_a) = \{x(t) \mid \exists c \in [-1, 1]\ \forall t : x(t) = c\}$$

and the reachable set is $Reach_t(S_a) = [-1, 1]$ for all $t \geq 0$.

Since both reachable sets are constant over time, it is easy to see that S_r is reachset conformant to S_a. However, all traces of S_a are constant traces, so none *of the sine traces of S_r is contained in S_a and S_r is not trace conformant to S_a. In Fig. 3 the reachable sets and some traces are shown. Although we use non-determinism only for the initial set, we also could use non-deterministic flow to design a similar example.*

Even though the system traces could be very different, we can nevertheless reason about safety properties of the system S_r with the abstract system S_a. A key point for applicability is an implementable conformance checking framework. Therefore, in the rest of the paper we are dealing with how to check reachset conformance.



(a) some traces (b) reachable sets

Figure 3: Systems S_r (solid, light gray) and S_a (dashed, dark gray) of Example 1.

4. REACHSET CONFORMANCE TESTING

In practice, it is hard to show that Eq. (1) holds for two systems S_r and S_a, because the system S_r is too complex and high-dimensional for formal methods, such as reachable set computations. However, falsification is possible by providing a counter-example proving the negation of Eq. (1):

$$\exists u(.) \in U(.) \exists t \geq 0 \, Reach_t(S_r, u(.)) \not\subseteq Reach_t(S_a, u(.)) \quad (3)$$

A practical approach is to use simulation runs or real data measurements as underapproximations of the reachable sets $Reach_t(S_r, u(.))$. Since neither the simulations nor the measurements provide exact data, we have to consider numerical errors or measurement errors. Therefore, we have to deal with an error bound ε and an approximation τ_T^ε of the true timed trace τ_T with

$$\max_{t \in T} \|\tau_T^\varepsilon(t) - \tau_T(t)\|_2 \leq \varepsilon. \quad (4)$$

Note that many other norms can be used, although the Euclidean norm is used here for the ease of presentation (cf. [17]). An overapproximation of the reachable set of S_a and an erroneous trace of S_r can be used to prove that S_r is not reachset conformant to S_a as depicted in Fig. 4.

PROPOSITION 3 (COUNTER-EXAMPLE). *A counter-example, falsifying the conformance relation $S_r \preceq_R S_a$ consists of*

1. *An input trajectory $u(.) \in U(.)$,*
2. *A point in time t,*
3. *An overapproximation $Reach_t^o(S_a, u(.))$ of the reachable set of S_a,*
4. *A sampled, erroneous trace $\tau_T^\varepsilon(.)$ of system S_r under input $u(.)$ with $t \in T$,*

where all elements x of the output space with $\|x - \tau_T^\varepsilon(t)\|_2 \leq \varepsilon$ are not contained in $Reach_t^o(S_a, u(.))$. This implies $S_r \not\preceq_R S_a$.

PROOF. Using Eq. (4), $\tau_T(t) = \tau(t)$ is also not contained in $Reach_t^o(S_a, u(.))$ which proves Eq. (3) and thus Eq. (1) cannot hold. □

An example is depicted in Fig. 5. If we check the erroneous sampled trace without considering the error, we get the points 2 and 3 as counter-examples. However, the true point 2 could possibly be contained in the box and we cannot be sure that it is not. By considering the error we are able to find the non-spurious counter-example point 3 only.

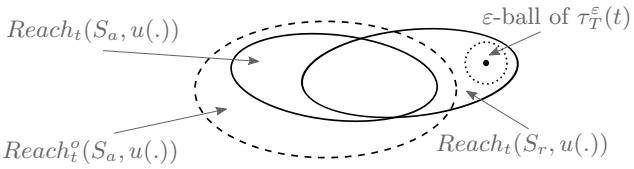


Figure 4: A counter-example falsifying the reachset conformance, because overapproximation (dashed) and ε-ball (dotted) are disjoint.

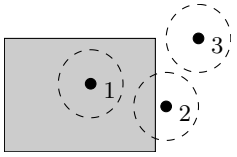


Figure 5: Error aware inclusion check: The true point is somewhere in the ball around the erroneous point, thus only point 3 is a non-spurious counter-example.

One advantage of this approach is that the sampled trace does not have to be a simulation. Erroneous measurements of the real system can be used also to falsify abstract models, which plays an important role for the applicability of model-based design. In the following we describe how to check for counter-examples.

4.1 Fixed input conformance testing

In this subsection, the conformance testing method as introduced by Althoff and Dolan [4] is described. In the following subsection we develop this method further and take the error bound ε for trace errors into account. This will lead to sound counter-examples and to more system pairs where the non-conformance can be proven.

The goal is to check if the non-conformance $S_r \not\preceq_R S_a$ can be shown by a counter-example for a given input $u(.)$. The test consists of 3 steps:

1. Compute an underapproximation $Reach_t^u(S_r, u(.))$ of the reachable states of S_r for any time t within a finite set T of points in time.

2. Compute an overapproximation $Reach_t^o(S_a, u(.))$ of the reachable set of S_a for each $t \in T$.

3. If $Reach_t^u(S_r, u(.)) \not\subseteq Reach_t^o(S_a, u(.))$ holds for any $t \in T$, at least one counter-example is found.

Rapidly-exploring random trees (RRTs) can be used to underapproximate $Reach_t(S_r, u(.))$, as described in [4]. They provide an efficient way of estimating the reachable set for complex systems by simulations and can also be used for black-box models, of which the dynamics are not known. As mentioned above, the first step could also be replaced by real measurements of a system.

The overapproximation of S_a can be efficiently computed using reachability analysis. Here, we consider the reachability tool CORA [3], where reachable set overapproximations are represented by zonotopes. Zonotopes are special convex set representations for efficient linear transformations and Minkowski addition (cf. [5]).

DEFINITION 3 (ZONOTOPE). *A n-dimensional zonotope Z in generator representation (G-representation) is the set*

$$Z = z(c, \langle g_1, \ldots, g_m \rangle) := \left\{ c + \sum_{i=1}^{m} \lambda_i g_i \middle| \lambda_i \in [-1, 1] \right\},$$

where $c \in \mathbb{R}^n$ is the center and $g_1, \ldots, g_m \in \mathbb{R}^n$ are the generators of Z.

Zonotopes are special, point symmetric polytopes:

DEFINITION 4 (POLYTOPE). *A n-dimensional polytope P in halfspace representation (H-representation) is the set*

$$P = p(H, k) := \{x \in \mathbb{R}^n \mid H \cdot x \leq k\}$$

with $H \in \mathbb{R}^{m \times n}$, $k \in \mathbb{R}^m$, also called a m-polytope.

The inclusion check by Althoff and Dolan [4] is done by abstracting from the zonotope and the samples to axis-aligned bounding boxes. Let v be the vector representing the box size in each dimension, then $v = 2 \sum_{i=1}^{m} |g_i|$ holds. Although the inclusion check is very fast, it introduces a very coarse overapproximation which leads to a conservative falsification result found with less counter-examples found. This problem actually increases with the number of output dimensions. Therefore we introduce a new approach for inclusion checking.

4.2 A configurable inclusion check with error-awareness

In this subsection we introduce a new inclusion check for points in a zonotope which leads to more counter-examples and a less conservative falsification result as later demonstrated in Sec. 6.3. We achieve this by reducing the error introduced by the transformation of the zonotope to an easily checkable representation. A useful approximation should give the possibility to configure the trade-off between accuracy and computational time, while providing an estimation of the approximation error. With the following inclusion check the trade-off can be freely adapted.

Reachability analysis for nonlinear dynamics with high accuracy needs a lot of generators, sometimes more than 1000. Because of scaling problems, methods that are only usable for a small number of generators are not directly applicable. Therefore, we are using overapproximations of zonotopes for the inclusion check.

In the following, let Z be a n-dimensional zonotope with center c and generators $g_1, \ldots, g_m$. For a polytope P in H-representation, a point x is contained in P, iff all inequalities $H \cdot x \leq k$ hold. This can be efficiently computed. Since zonotopes are special polytopes, they can be transformed to H-representation by using one inequality for every facet. However, Althoff et al. [5] showed that for a zonotope in dimension n with m independent generators the number of facets is $2 \begin{pmatrix} m \\ n-1 \end{pmatrix}$. Hence, the exact transformation approach does not scale, especially for $m \geq 1000$ and $n \geq 3$. However, by using support functions, described by Girard et al. [11, 17], the zonotope can be tightly overapproximated.

DEFINITION 5 (SUPPORT FUNCTION). *Let a zonotope Z be given. Then for $d \in \mathbb{R}^n$ the support function of Z is*

$$\rho_Z(d) = \max_{x \in Z} d^T \cdot x = d^T \cdot c + \sum_{i=1}^{m} |d^T \cdot g_i|.$$

Since the resulting overapproximation is a polytope, the H-representation can be used for inclusion checking. The zonotope is point symmetric to its center. Therefore, the directions d and $-d$ can be easily checked together. Hence, the inclusion in a $2l$-polytope can be checked with l directions.

PROPOSITION 4 (OVERAPPROXIMATION). *Let a finite set of directions $D \subset \mathbb{R}^n$ and a zonotope Z be given. Then*

$$Z \subseteq \bigcap_{d \in D} H_d$$

holds, where H_d are the halfspaces

$$H_d = \{x \in \mathbb{R}^n \mid d^T \cdot x \leq \rho_Z(d)\}.$$

A point $x \in \mathbb{R}^n$ is contained in $H_d \cap H_{-d}$, iff

$$|d^T \cdot x - d^T \cdot c| \leq \rho_Z(d) - d^T \cdot c$$

holds [10].

Using this polytope, the inclusion can be checked for the approximation τ_T^ε using the error bound ε as shown next.

PROPOSITION 5 (INCLUSION CHECK). *Let τ_T^ε and an overapproximation Z of the reachable set $Reach_t(S_a, u(.))$ be given. The inequality*

$$|d^T \cdot \tau_T^\varepsilon(t) - d^T \cdot c| > \rho_Z(d) - d^T \cdot c + \varepsilon \|d\|_2 \qquad (5)$$

for any d implies that the real state $\tau(t)$ is not contained in $Reach_t(S_a, u(.))$.

PROOF. If the center c of the zonotope is not the origin, we can translate the zonotope and the point with $-c$. Therefore, without loss of generality $c = 0$ and $\|d\|_2 = 1$ holds. Let us assume the real state $\tau(t)$ is contained in the zonotope Z and Eq. (5) holds. This leads to the equation

$$|d^T \cdot \tau_T^\varepsilon(t)| > \rho_Z(d) + \varepsilon \geq |d^T \cdot \tau(t)| + \varepsilon. \qquad (6)$$

However Eq. (4) and the triangle inequality lead to

$$|d^T \cdot \tau_T^\varepsilon(t)| \leq |d^T \cdot \tau(t)| + |d^T \cdot (\tau_T^\varepsilon(t) - \tau(t))| \leq |d^T \cdot \tau(t)| + \varepsilon,$$

which is a contradiction to Eq. (6) □

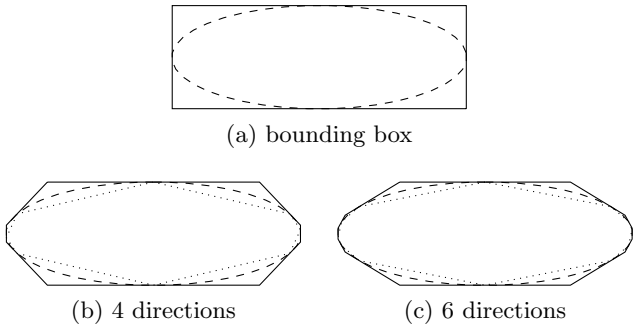
The directions remain as free parameters, so that we can tune the accuracy and computational cost with their selection. For example, when selecting the directions e_i, where the e_i are the canonical basis vectors, the aforementioned box overapproximation used by Althoff and Dolan [4] is obtained. Hence, it is a special case of the presented method.

Since a priori there is no knowledge about the zonotope generators, the selected directions should be evenly distributed over the space of possible directions or evenly distributed over one halfspace of $\mathbb{R}^n$ considering the symmetry of the zonotope. While optimization-based direction generation methods iteratively improve their solution, explicit methods have the advantage of directly generating good directions. In 2 dimensions, l evenly distributed directions $d_1, \ldots, d_l$ are

$$d_i = \left(\cos\left(\frac{i\pi}{l}\right), \sin\left(\frac{i\pi}{l}\right) \right)^T. \qquad (7)$$

In 3 dimensions, the Fibonacci lattice can be used, as described by González [13]. The directions $d_1, \ldots, d_l$ are generated via

$$d_i = (\sin(lat_i)\cos(lon_i), \sin(lat_i)\sin(lon_i), \cos(lat_i))^T, \quad (8)$$



(a) bounding box

(b) 4 directions (c) 6 directions

Figure 6: Example of overapproximations (solid) and underapproximations (dotted) of a zonotope (dashed).

where the angles are

$$lat_i = \arcsin\left(\frac{2(i-1)}{2l-1}\right) \text{ and } lon_i = \pi(i-1)(\sqrt{5}-1)$$

As far as we know, there is no applicable explicit method in higher dimensions, hence in this case we use an optimization-based direction generation method. Since we are generating evenly distributed directions in a preprocessing step and use the same set for every inclusion check over time, the computational load of the direction generation is independent of the number of inclusion checks.

To generate good directions by optimization, a simple method by Frehse et al. [9] is used. First, m directions are randomly generated. Then, a direction d is randomly generated and the nearest direction is replaced by d if the distribution of the other directions with d is more uniform. This can be done as long as a termination condition on the uniformity is not fulfilled.

EXAMPLE 2 (ELLIPSE). *In Fig. 6, an 2D example is presented. The considered zonotope* $Z = z(0, \langle g_1, \ldots, g_{20}\rangle)$ *has 20 generators*

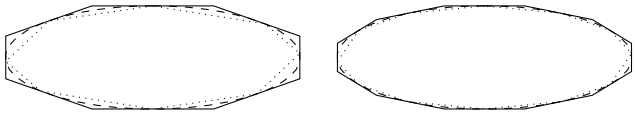
$$g_i = \left(3\sin\left(\frac{\pi i}{20}\right), \cos\left(\frac{\pi i}{20}\right)\right)^T$$

and is very close to an ellipse. The overapproximations are generated via Eq. (7). While the box overapproximation is very coarse, the configurable approximation consisting of 4 respectively 6 directions that approximate the zonotope more tightly. With more or less directions the accuracy and computational time can be tuned. Note that with l *directions the used overapproximation is a* $2l$*-polytope.*

In Fig. 6 one can see that the overapproximation is not very tight if different dimensions have different scales. Therefore we normalize the directions according to the axis-aligned bounding box of the zonotope to produce a tighter overapproximation. Let $W = diag(v_1, \ldots, v_n)$ be the diagonal matrix consisting of the box size of each dimension of the bounding box. Then a direction d is normalized to

$$d' := W^{-1}d. \tag{9}$$

EXAMPLE 3. *Considering the normalization with* $W = diag(3, 1)$ *for the ellipse of Example 2, the approximation is tighter as depicted in Fig. 7.*



(a) 4 directions (b) 6 directions

Figure 7: Example of overapproximations (solid) and underapproximations (dotted) of a zonotope (dashed) with normalized directions.

With Eq. (9) and (5) the inclusion check for a overapproximation Z, a set of directions D, and a set of points M with maximum error ε, can be implemented. If a counterexample is found, it will be returned. Otherwise, false will be returned. For a practical example and comparison of the introduced method, see Sec. 6.3.

4.3 Quality of the zonotope overapproximation

Since overapproximations of zonotopes are used for the inclusion check, the non-inclusion of some points cannot be seen. Therefore, we want to quantify the error introduced by the overapproximation, e. g. to decide if more directions are needed. Althoff et al. [5] introduced a relative quality measure $\Theta := \sqrt[n]{\frac{vol(W\cdot Z^o)}{vol(W\cdot Z)}}$ for the overapproximation Z^o of a n-dimensional zonotope Z. The volume vol in $\mathbb{R}^n$ is defined as the Lebesgue measure and the matrix W is a normalization matrix. Since the exact volume of the zonotope Z cannot be computed easily, the quality measure is not directly applicable here. Therefore we present a method to bound the overapproximation error.

Every support function $\rho_Z(d)$ comes with an extremal point

$$p_d := c + \sum_{i=1}^{m} sign(d^T \cdot g_i)g_i$$

of the zonotope and the convex hull of these points forms an underapproximation of the zonotope, as shown in [12]. Thus, this can be used to get a bound for the approximation error.

PROPOSITION 6. *The measure* $\Theta_* := \sqrt[n]{\frac{vol(W\cdot Z^o)}{vol(W\cdot Z^u)}}$ *with* $Z^u := convexhull(\{p_d \mid d \in D\})$ *is an upper bound for the relative error and* $vol(W \cdot Z^o) \leq \Theta_*^n vol(W \cdot Z)$ *holds.*

Since the approximating polytopes have less facets than the original zonotope, it is faster to compute the volumes.

EXAMPLE 4. *In the 2-dimensional Example 2 the relative size of the configurable approximation can be bounded by* $\Theta_* = 1.161$ *respectively* $\Theta_* = 1.098$*, whereas with normalization in Example 3 the bounds are* $\Theta_* = 1.079$ *respectively* $\Theta_* = 1.036$.

5. INPUT SELECTION AND COVERAGE

In the previous section we describe how to check conformance for a given input. However, we have not yet discussed how to select the inputs. This is an important step, because when inputs are selected such that they nearly generate the same output, the conformance check might miss behaviours which are non-conformant. Hence, we are interested in selecting the inputs that produce different outputs. Furthermore, we are interested in a small number of

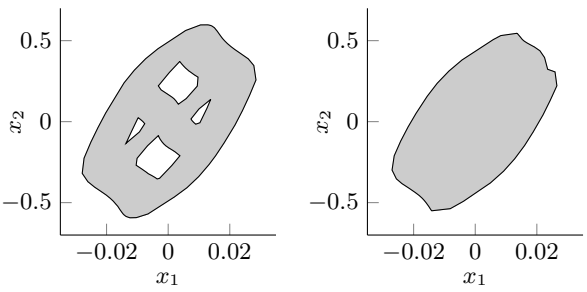


Figure 8: Example of the covered state space (gray) for different inputs.

test cases, because simulation results of a complex system and real measurements are costly to obtain. Therefore we present a method to reduce the number of test cases from a generated finite test set.

Since we are focusing on non-deterministic models for S_a in this paper, we assume that the reachable sets have a volume greater than zero. Otherwise, one has to consider only the spanned dimensions. Due to non-determinism, these dimensions are time-invariant and can be selected offline.

To speed up the process, we introduce a method to preselect input trajectories without computing the output of both systems and without conformance checking. In literature, different methods for input sampling have been introduced, such as Monte Carlo sampling. Another method presented by Dang [7] generates input samples based on rapidly-exploring random trees such that the reachable space is approximately covered. However, not all of the generated inputs can be performed on the original model, because this is too costly. Assuming a finite set of test cases U_1 is generated by the aforementioned methods, we present a method to select an input subset U_2. The method compares different inputs by comparing the reachsets of S_a under the inputs. Therefore, input trajectories whose output can be also achieved by another input trajectory with non-determinism can be removed. The assumption is that inputs reaching the same states on system S_a are less interesting for conformance testing than other inputs that reach new states for S_a.

Hence, we are interested in an input set that covers the reachable set of the system S_a. Although a priori the overall reachable set is not known, we are able to use the reachable sets $Reach_t(S_a, u(.))$ to define a coverage measure and select a relevant subset U_2 of an input set U_1.

Definition 6 (Coverage measure). *Let a system S and an input set U be given. Then the covered state space is*

$$Reach(S,U) = \bigcup_{u(.)\in U} \bigcup_{t\geq 0} Reach_t(S,u(.))$$

and a coverage measure is $vol(W \cdot Reach(S,U))$, where W is a normalization matrix similar to the one in Eq. (9).

Since exact reachable set comparison and volume computation is typically not possible for nonlinear dynamics and complex geometric sets, we evaluate it in an overapproximative way, denoted by $Reach^o$. This can be used as a heuristic to iteratively pick the input that increases the state space coverage the most. For example in Fig. 8 the right input

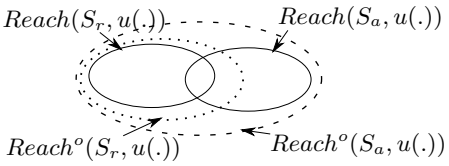


Figure 9: Comparison of the overapproximation does not lead to a non-spurious statement of the reachable sets.

covers a bigger part of the state space than the left one and thus should be selected. The covered space of the selected inputs is compared to the initial covered space by the input set U_1.

Definition 7 (Relative coverage). *Let two sets of inputs U_1 and U_2 with $U_2 \subseteq U_1$ for a system S be given. Then the relative covered state space is*

$$rcov(S,U_1,U_2) = \frac{vol_2}{vol_1},$$

where the vol_i are computed as the volumes of $Reach(S,U_i)$ as defined in Def. 6.

A greedy input selection algorithm can be implemented by iteratively choosing the input which increases the coverage measure the most. With a given parameter ϵ for the relative coverage needed, we can adapt the trade-off between reaching the whole covered state space of U_1 and the size of the input set U_2. If the dimensions have different scales, a normalization for the volume computation could be applied via W to get a better representation of small scale dimensions.

If we compare the overapproximation for two input trajectories, we cannot formally argue about inclusions. As depicted in Fig. 9, the overapprox. $Reach^o(S_r, u(.))$ of system S_r is enclosed by the overapproximation $Reach^o(S_a, u(.))$, although the covered state space $Reach(S_r, u(.))$ is not contained in $Reach(S_a, u(.))$. To prove that $Reach(S_r, u(.))$ is contained in $Reach(S_a, u(.))$ we would need an underapproximation. Since, as far as the authors know, tools that compute tight underapproximations do not exist yet, especially for nonlinear dynamics, $Reach^o$ is used as a heuristic only and does not give formal bounds. Often, the overapproximation is relatively close to the exact reachable set and therefore the heuristic is also close to the theoretically intended measure.

6. EXPERIMENTS

In this section the presented methods are evaluated on an example from the domain of autonomous driving. We first describe which models we use, how the inputs are selected, and then how the directions are chosen for conformance testing. Finally, we show and discuss the numerical results.

6.1 Models

We consider the setup and the two models used by Althoff and Dolan [4] with the friction coefficient $\mu = 0.9$. The systems S_r and S_a are models of an autonomous car that follows a planned trajectory. The model S_a is a 6-dimensional continuous bicycle model which models: The 2-dimensional position of the center of mass x and y, the heading angle ψ, the yaw rate $\dot{\psi}$, the velocity v, and the slip angle β. The model S_r published by Allen et al. [2, Appendix A] is a more

complex model with 28 continuous variables and bounded actuators, thus it has some simple hybrid behaviour. The output of S_r is the projection of its states onto the state space of S_a and thus 6-dimensional. The non-determinism of both systems models sensor inaccuracies, such as disturbances in the position perception.

Since the bicycle model has simplified dynamics and especially simplifies the estimation of slip angle β and the friction influence on v, additional non-determinism of the bicycle model flow is introduced for β and v

$$\dot{\beta} \in F_\beta(.) + [-d_\beta, d_\beta], \quad \dot{v} \in F_v(.) + [-2d_v, 0],$$

where $F_\beta(.)$ and $F_v(.)$ are the differential equations of β and v without non-determinism and d_β, d_v are the parametric bounds of the non-determinism.

The input space of the models consists of trajectories of the x- and y-position, the heading angle, the yaw rate, and the velocity $(x, y, \psi, \dot{\psi}, v)$ with initial state $(0, 0, 0, 0, 15)$ and evolutions bounded by

$$\|(a_x, a_y)^T\|_2 < 7[m/s^2] \text{ and } \|(\dot{a}_x, \dot{a}_y)^T\|_2 < 50[m/s^3]$$

where $a_\cdot$ is the x- respectively y-acceleration of the vehicle. Furthermore, we fix the velocity to $15[m/s]$ for simplicity of presentation.

6.2 Input selection

The input space is randomly sampled by input trajectories, where the lateral acceleration is constant for 0.2 seconds respectively approaches the choosen acceleration with maximum acceleration rate. Therefore, we get a set of inputs U_1 with 5000 driving maneuvers of 2 seconds length.

Since vehicle dynamics are invariant with respect to position x, y and heading angle ψ, we do not consider these state variables for the coverage measure and thus project the reachable set to the other state variables. The method described in Sec. 5 is used to choose the set of inputs U_2 with $\varepsilon = 0.96$. The resulting four input trajectories are compared to other sets of inputs that are random selections of the same size. Since the coverage computation and the input selection took only slightly more time compared to one full inclusion check, it successfully speeded up the conformance test.

6.3 Inclusion check

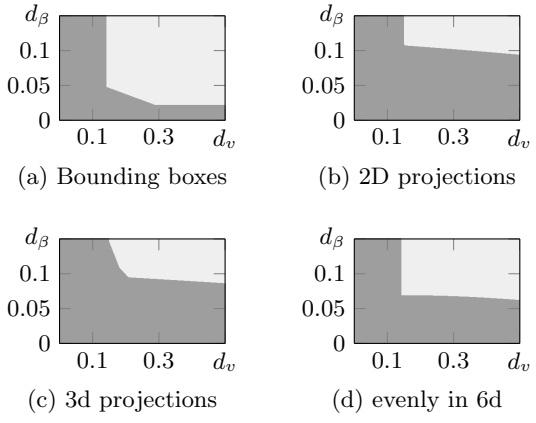
We discretized the time into around 3500 points. For every point in time, the RRT-algorithm generated 70 samples from the reachable set of S_r and the zonotope overapproximation of the reachable set of S_a is generated. The inclusion check with normalized directions is done as described in Sec. 4 using 4 different direction selection methods: (i) Axis-aligned bounding boxes, (ii) overapproximations on every 2D projection on two state dimensions, (iii) overapproximations on every 3D projection on three state dimensions, (iv) overapproximation with evenly distributed directions in 6D. For the 2D and 3D projections we consider all possible projections and select the directions according to Eq. (7) and Eq. (8) in Sec. 4. The evenly distributed directions in 6D are obtained by the optimization method, described in Sec. 4. Consider that we use the same amount of directions for all methods, except for the box check, as shown in Table 1.

6.4 Results

The results for the four directions selections and 257 sampled values for each parameter d_β and d_v of the abstract system S_a are visualized for one input trajectory in Fig. 10. Parameter combinations are colored dark gray if the conformance test found a counter-example. As one can see, our direction choices lead to more parameter combinations, where $S_r \not\preceq_R S_a$ can be proven. Clearly, the choice of the directions directly influences the falsification result.

boxes	2D proj.	3D proj.	evenly in 6D
6	$15 \cdot 40$	$20 \cdot 30$	600

Table 1: Number of directions used for each method.



(a) Bounding boxes (b) 2D projections

(c) 3d projections (d) evenly in 6d

Figure 10: Inclusion check for different parameter combinations. Dark gray: $S_r \not\preceq_R S_a$ proven, light gray: no counter-example found.

The dark gray area in Fig. 10 can be used to compare the falsification results of different set of inputs. Therefore we use the ratio of the dark gray area to the whole considered area as a falsification measure. A high falsification measure states that the method is able to falsify many parameter combinations, which is good for a falsification method. We compare the ratio for our selected set of inputs against several randomly chosen ones. The set of inputs U_2 gives a good falsification result for all four methods, see Fig. 11. However, there is one input set that gives a better result than our selected set of inputs. This cannot be shown by the already existing bounding box check, but with our new method. Since there are no formal guarantees that our algorithm picks the best set of inputs, some inputs can lead to better results depending on the dynamics of the real model S_r that are not used for selection. Note that in our case we get similar results for evenly distributed directions in 6D, 2D and 3D projections. Possibly, this is due to the considered systems dynamics and the relation of the state variables therein. Depending on the resulting shape of the reachable set, it can be more accurate to check projections rather than the exact set giving a fixed number of directions. Nevertheless, in this particular example there is significantly more falsification possible with our new approach.

7. RELATED WORK

While we focus on reachset conformance in this paper, we also relate it to trace conformance (cf. Sec. 3). Therefore we briefly discuss trace conformance for a comprehensive overview. There are various conformance relations for differ-

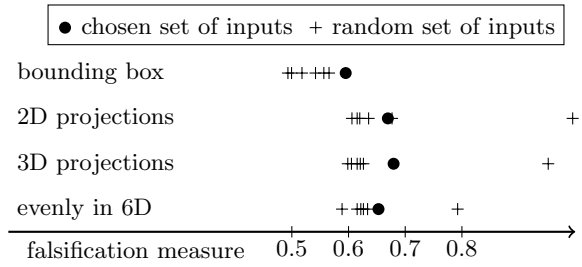


Figure 11: Falsification measure for different input sets.

ent types of models. The IO conformance (IOCO) is a formal approach to conformance testing of purely discrete models (labelled transition systems) by Tretmans [22]. IOCO has been extended to timed systems with subtle differences, see work by Schmaltz et al. for an overview [20]. Note, in the overview [20] the same wording "reachable set" is used, although input and output actions are considered together as transitions, leading to a different meaning. IOCO was also extended to hybrid systems conformance by van Osch [23] based on hybrid transition systems. A similar notion of hybrid conformance based on hybrid automata is described by Dang [7]. Approximate simulation relations are used by Tabuada [21] to verify models based on abstractions. Abbas et al. [1] use (τ, ε)-conformance, where the traces of the two systems only have to be close to each other. They prove that if their conformance relation $S_r \preceq_{(\tau,\varepsilon)} S_a$ holds, only a transformed version of the properties of S_a holds on S_r. A comparison between Hybrid Input-Output Conformance, Approximate Simulation, and (τ, ε)-conformance is done by Khakpour and Mousavi [16]. All of the above mentioned conformance relations are basically trace conformance relations.

There are different strategies in literature for overapproximating a zonotope with a simplified representation. Girard et al. [10] and Althoff et al. [5] present methods to reduce the number of generators of a zonotope. While the reduction to a small number of generators helps to scale the inclusion check, there is a significant penalty in accuracy of the inclusion check as the simplified zonotope is close to the box approximation. Girard [12] use zonotope approximations to check if the zonotope intersects with a guard of a hybrid automaton. Guibas et al. [14] describe an exact inclusion check for zonotopes that is limited to 3 dimensions only. The inclusion check presented in this paper is based on support function that are used e. g. in SpaceEx for reachable set computations [9]. While SpaceEx could also be leveraged for our approach, it is restricted to affine hybrid system models. Similarly, C2E2 [8] could be possibly leveraged for conformance testing, however it requires to annotate the model with certificates called discrepancy functions. If these certificates are given, Mitra provide a conformance checking procedure for continuous systems without inputs that particularly focus on security [19]. In this work we consider CORA [3] for the reachable set computation for the following reasons: (*i*) it supports non-linear hybrid systems, (*ii*) it allows us to easily incorporate our new reachset conformance, (*iii*) it provides a useful zonotope representation for reachable sets and (*iv*) allows us to compare our results to previous conformance testing on the autonomous vehicle models by Althoff and Dolan [4]. Kanade et al. [15] have done a reachable set underapproximation of Simulink models restricted to linear transformations. However, since their method takes a trace and builds a reachset around it, it does not consider different discrete behaviour. Generally, for verification purposes we would also need set-based underapproximation techniques for non-linear hybrid systems, that are still missing. Backward reachability for example is not usable because of ill-conditioning, as outlined by Mitchell [18].

For test generation of discrete systems there are several methods for test generation, such as transition coverage for finite automata. However, these methods do not work well for hybrid systems because they do not consider any continuous flow. A test generation method is proposed by van Osch [23] that has a non-deterministic selection process. Since it has no selection heuristic, it does not use knowledge about the system in contrast to our method. A RRT-based test generation process was introduced by Branicki et al. [6]. Dang [7] further developed the approach by using a statistical measure called star discrepancy to guide the simulations to unreached parts of the state space. However, there are typically too many inputs in the resulting input set to apply them all on a complex model.

8. CONCLUSIONS

We introduce the formal definition of reachset conformance and prove the transference of safety properties. Since the reachset conformance is weaker than trace conformance it can be used to relate more systems and therefore properties transfer between more systems. We present a formal reachset conformance testing, which is based on reachable set computations and overapproximations with support functions and considers the error of simulation runs or real measurements. The trade-off between accuracy and computational load can be tuned by an appropriate choice of the directions for the overapproximations. We introduce an input selection algorithm to reduce the size of an input set, generated by existing sampling methods. It uses a coverage measure based on the reachable sets of the abstract system. The example shows that the selected inputs are reasonable and that the conformance testing method can falsify more relations than the state of the art.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers as well as Christoph Gladisch, Thomas Heinz and Christian Heinzemann for their valuable comments and suggestions to improve the quality of the paper. Jens Oehlerking, Matthias Woehrle and Matthias Althoff gratefully acknowledge financial support by the European Commission project UnCoVerCPS under grant number 643921.

9. REFERENCES

[1] H. Abbas, H. D. Mittelmann, and G. E. Fainekos. Formal property verification in a conformance testing framework. In *ACM/IEEE MEMOCODE*, pages 155–164, 2014.

[2] R. W. Allen, H. T. Szostak, D. H. Klyde, T. J. Rosenthal, and K. J. Owens. Vehicle dynamic stability and rollover. Technical report, U.S. Department of Transportation, Final Report DOT HS 807 956, 1992.

[3] M. Althoff. An introduction to CORA 2015. In *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 120–151, 2015.

[4] M. Althoff and J. M. Dolan. Reachability computation of low-order models for the safety verification of high-order road vehicle models. In *American Control Conference*, pages 3559–3566. IEEE, 2012.

[5] M. Althoff, O. Stursberg, and M. Buss. Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes. *Nonlinear Analysis: Hybrid Systems*, 4(2):233–249, 2010.

[6] M. S. Branicky, M. M. Curtiss, J. Levine, and S. B. Morgan. RRTs for nonlinear, discrete, and hybrid planning and control. In *Decision and Control*, volume 1, pages 657–663. IEEE, 2003.

[7] T. Dang. Model-based testing of hybrid systems. In *Model-Based Testing for Embedded Systems*, chapter 14, pages 383–424. CRC Press, Inc., 2011.

[8] P. S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok. C2E2: A verification tool for Stateflow models. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 68–82. 2015.

[9] G. Frehse, C. L. Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable verification of hybrid systems. In *Computer Aided Verification*, pages 379–395, 2011.

[10] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, pages 291–305. 2005.

[11] A. Girard and C. L. Guernic. Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Hybrid Systems: Computation and Control*, pages 215–228. 2008.

[12] A. Girard, C. L. Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In *Hybrid Systems: Computation and Control*, pages 257–271. 2006.

[13] Á. González. Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical Geosciences*, 42(1):49–64, 2010.

[14] L. J. Guibas, A. Nguyen, and L. Zhang. Zonotopes as bounding volumes. In *Proc. of the ACM-SIAM Symposium on Discrete Algorithms*, pages 803–812, 2003.

[15] A. Kanade, R. Alur, F. Ivancic, S. Ramesh, S. Sankaranarayanan, and K. C. Shashidhar. Generating and analyzing symbolic traces of Simulink/Stateflow models. In *Computer Aided Verification*, pages 430–445, 2009.

[16] N. Khakpour and M. R. Mousavi. Notions of Conformance Testing for Cyber-Physical Systems: Overview and Roadmap. In *Int. Conf. on Concurrency Theory*, volume 42 of *LIPIcs*, pages 18–40, 2015.

[17] C. Le Guernic, A. Girard, C. L. Guernic, and A. Girard. Reachability analysis of hybrid systems using support functions. In *Computer Aided Verification*, pages 540–554, 2009.

[18] I. M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In *Hybrid Systems: Computation and Control*, pages 428–443. 2007.

[19] S. Mitra. Proving abstractions of dynamical systems through numerical simulations. In *Proc. of the Symposium and Bootcamp on the Science of Security*, page 12, 2014.

[20] J. Schmaltz and J. Tretmans. On conformance testing for timed systems. In *Formal Modeling and Analysis of Timed Systems*, pages 250–264. 2008.

[21] P. Tabuada. *Verification and Control of Hybrid Systems - A Symbolic Approach*. Springer, 2009.

[22] G. J. Tretmans. *A formal approach to conformance testing*. PhD thesis, Universiteit Twente, 1992.

[23] M. P. W. J. van Osch. *Automated model-based testing of hybrid systems*. PhD thesis, Eindhoven University of Technology, 2009.