mathematics

MDPI

Article

Reachset Conformance and Automatic Model Adaptation for Hybrid Systems

Hendrik Roehm [1,*], Alexander Rausch [2] and Matthias Althoff [1]

[1] Department of Informatics, Technical University of Munich, 85748 Garching, Germany ; althoff@in.tum.de
[2] Corporate Research, Robert Bosch GmbH, 71272 Renningen, Germany; alexander.rausch3@de.bosch.com
* Correspondence: hendrik@roehm.ws

Abstract: Model-based verification uses a model to reason about the correctness of a real system. This requires the model and the system to be conformant, such that verification results on the model can be transfered to the real system. Especially for hybrid systems, which combine discrete and continuous behavior, defining and checking conformance is a difficult task. In this work, we present reachset conformance for hybrid systems that transfers safety properties from a model to the real system. We show how a model can be adapted to be conformant to measurements of a real system and demonstrate this for a real autonomous vehicle. The obtained reachset conformant model can be used for the verification of safety-critical properties, such as collision avoidance.

Keywords: conformance; reachability analysis; formal verification; hybrid systems; automated vehicles; uncertain systems

MSC: 37M10

check for updates

Citation: Roehm, H.; Rausch, A.; Althoff, M. Reachset Conformance and Automatic Model Adaptation for Hybrid Systems. *Mathematics* **2022**, *10*, 3567. https://doi.org/10.3390/math10193567

Academic Editor: Julien Alexandre dit Sandretto

Received: 25 August 2022
Accepted: 26 September 2022
Published: 29 September 2022

1. Introduction

The amount of technical systems operating autonomously and without human interference is continuously growing. It is of utmost importance to demonstrate that such autonomous systems cannot harm people, cause damage, or breach any other important specification. Autonomous vehicles and moving robots are examples of such safety-critical applications, where the discrete and continuous aspects are tightly intertwined—these systems are often referred to as hybrid systems.

Model-based formal verification is an important approach toward safer systems. In particular, reachability analysis computes the set of reachable states of a model and can be used to check whether unsafe states are possible [1,2]. To include different possible behaviors of the real system, verification models for reachability analysis are non-deterministic. This means that at each point in time, there might be multiple evolutions of the model, and one has to reason about all of them. The basic assumption of model-based verification is that the model is related to the system in a way that the safety of the system can be implied if the safety of the model has been shown. We call a relation between a system and a model *conformance relation* and argue that it should also be defined formally. Otherwise, one cannot be sure that the used conformance relation allows to transfer safety properties from the model to the real system, which would make the formal verification effort useless. Hence, the conformance relation connects the formal world of reasoning with the real world.

A major challenge of a verification model is that it simultaneously has to be amenable to verification and conformance. This is challenging because (i) a model with significant non-determinism is conformant, but it might have too many reachable states to verify properties (model 1 in Figure 1) and (ii) a model with insignificant non-determinism is amenable for verification, but it might not be conformant to the real system because some states cannot be produced by the model (model 4 in Figure 1). Between these two extremes are the most useful models, amenable for both verification and conformance (model 2 and model 3 in Figure 1). We argue that an optimal model has just enough non-determinism such that reachset conformance holds to sustain the most freedom possible for verification. A method is needed to build verification models maximizing verification capabilities while ensuring conformance.

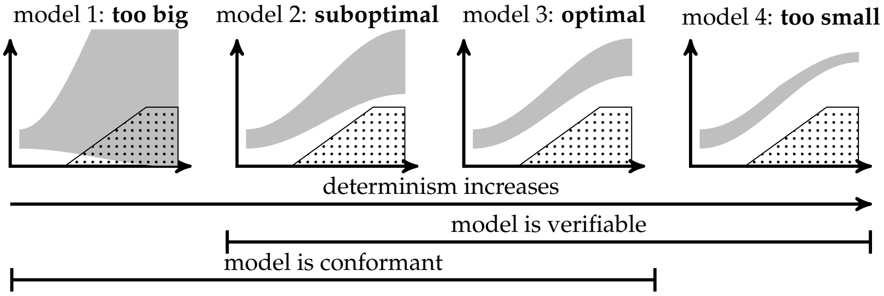


Figure 1. The reachable states (gray area) of several verification models as well as the unsafe states (dotted area) are shown. For increasing determinism, the set of reachable states is becoming smaller. When the reachable set is too small to contain all possible states of the real system, it is no longer conformant. Also, when the reachable is too big, it intersects with the unsafe state, and thus, it cannot be used for successful verification. The optimal model has the most determinism while being conformant.

In a previous paper [3], we introduced *reachset conformance* and showed how to test it for a given (hybrid-system) model M and measurements of a (real) system S. Here, we extend these results in several aspects:

Natural choice: We show that safety properties can be transfered from M to S exactly in the case when reachset conformance between M and S holds. Therefore, reachset conformance is the natural choice to transfer safety properties.

Quantified reachset conformance check: A robustness measure is introduced, which is based on the distance of a point to the boundary of a reachable set. In our setting, reachable sets are represented as zonotopes, and as a result, exclusion can be checked using linear programming techniques. This is computed for the measured data of the real system of some input and the output of the verification model for the same input, cf. Figure 2.

Model adaptation: We show how to automatically adapt the non-determinism of a model M, such that M becomes reachset-conformant to S. This is computed by identifying bounds on non-deterministic parameters. The bounds are optimized using Bayesian optimization to minimize the non-determinism while being conformant, Figure 1. In addition to building a reachset-conformant model, the method maximizes the verification capabilities of the model, cf. model 3 in Figure 2. Thus, our method helps to overcome the burden of building a formal verification model.

Autonomous vehicle application: We apply our methods to a real automated vehicle and construct a verification model of the vehicle. Measured driving data of the automated vehicle were recorded, and our model adaptation was applied to identify the non-determinism of the verification model such that the model is reachset-conformant to the automated vehicle. Our verification model is amenable to verification, showing that our approach is applicable to the highly relevant use case of autonomous vehicles. This is the first work showing reachset conformance for a real (autonomous) vehicle.

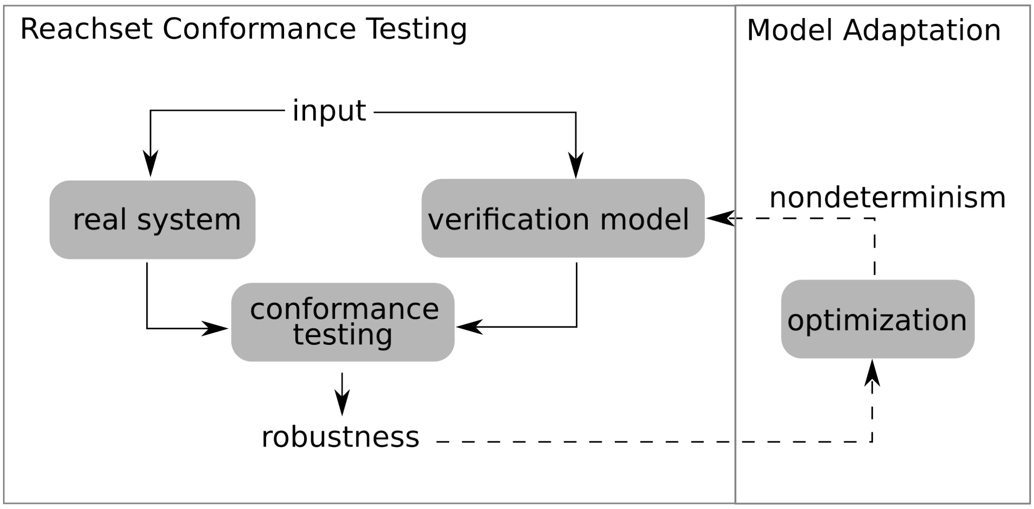


Figure 2. Overview of the reachset conformance testing and model adaptation methods.

The paper is structured as follows: First, we discuss related work on conformance and verification in Section 2. In Section 3, we present the underlying formalism of hybrid automata and other preliminaries. In Section 4, we present reachset conformance, prove that it is necessary and sufficient to transfer safety properties, and compare it to trace conformance, which is discussed in [4]. In Section 5, we present a testing method for reachset conformance. In Section 6, we introduce our model adaptation algorithm. In Section 7, we apply the presented techniques to a real vehicle and build a reachset-conformant vehicle model. Finally, we give some conclusions and future directions.

2. Related Work

2.1. Verification

An overview of safety verification for hybrid systems is provided by Guéguen et al. [1]. Schupp et al. [5] give an overview of methods for the reachability analysis of hybrid systems and discuss their challenges with respect to verification. In this work, we focus on reachability-based verification techniques. In the context of our application example *autonomous vehicles*, the reachability analysis tool CORA [6] has been used for the verification of cooperative and non-cooperative maneuvers of autonomous vehicles [7]. The reachability analysis computations are fast enough to allow online verification while the vehicle is driving [7].

2.2. Conformance

A variety of conformance relations have been defined. For brevity, we are only mentioning the most important ones here and refer to the survey by Roehm et al. [8] for a detailed overview of different conformance relations. Since properties are typically specified on output traces, one important notion of conformance is the *trace conformance* for hybrid automata [4] and similarly *hybrid input–output conformance* [9]. It requires that all possible output traces of one system S are also output traces of the other system M. Trace conformance reflects the conventional notion of conformance of discrete automata where traces of one system also have to be traces of the other one [10]. When trace conformance holds, universally quantified properties are transfered, such as metric temporal logic formulas [11].

Trace conformance has been generalized to approximate versions, which do not require the traces of one system to be included in the other but allow some deviation. The (τ, ε)-closeness [12] allows for some value-differences as well as time-shifts. (τ, ε)-closeness testing is performed by using a robustness value as a heuristic to guide the testing to non-conformant behavior [13]. ε-δ-similarity is a similar notion to (τ, ε)-closeness but does not allow local time disorder in the comparison of two traces [14]. The ε-Skorokhod conformance [15] uses the Skorokhod metric to quantify the distance between traces. The Skorokhod distance can be also computed between reachpipes [16], which are for instance traces with an ε-ball around them. One problem is that approximate relations do not transfer properties directly, but the property alters on transference [11,15].

So far, we only reviewed conformance relations applied to the output space. On the contrary, there are a variety of other conformance relations relating states of the systems which are called simulation relations [17–22]. The basic idea is that states of both systems are related such that evolutions from any state of one system can be mirrored by evolutions of a related state of the other system. There also exist approximative versions of simulation relations, which require the states to just be approximately similar [23–25]. On one hand, simulation relations transfer more properties compared to trace conformance. On the other hand, they require the systems to be more similar to each other than trace conformance. Since simulation relations require knowledge of the states, real systems cannot be considered since their exact state is unaccessible, except when all states can be measured. Hence, we will focus on comparing the reachset conformance relation to trace conformance (cf. Section 4). For a discussion between simulation relation and reachset conformance, we refer to the survey by Roehm et al. [8].

While we have introduced *reachset conformance* in 2016 [3], there is already some work in the direction of identification of non-determinism for reachset conformance. Liu and Althoff [26] have published a method to identify non-determinism for dynamical systems to be reachset conformant and applied it to the reachset conformance of robots. Kochdumper et al. [27] have shown a synthesis algorithm for the reachset conformance of linear hybrid systems leveraging the internal structure of the hybrid system. Furthermore, methods to bound additive errors for discrete-time dynamical models have been published [28,29]. Contrary to these methods, our method does not require any special knowledge on the hybrid system and is not restricted to a subclass of hybrid systems.

There exist methods for finding the parameters of a deterministic hybrid system to approximate measured data [30]. Our work differ in that we are identifying the non-determinism needed to include measurements in a non-deterministic model.

3. Preliminaries

In this paper, we use *hybrid automata* as a modeling formalism. A hybrid automaton can be seen as a finite automaton whose discrete states are annotated with differential inclusions that define the non-deterministic evolution of the continuous states [31]. An overview of definitions of hybrid automata and their differences has been presented by Frehse [17]. In our work, a (non-deterministic) hybrid automaton H consists of

- A finite set of locations $Q \subset \mathbb{N}$;
- A continuous state space $X \subseteq \mathbb{R}^n$;
- An initial set $I_H \subseteq Q \times X$;
- A continuous input space $U \subseteq \mathbb{R}^m$;
- A flow function $F_H : Q \times X \times U \to \mathcal{P}(X)$, where $\mathcal{P}(X)$ is the power set of X;
- An invariant set $inv(q) \subseteq X$ for each location q;
- A set of discrete transitions $\mathcal{T} \subseteq Q \times Q$;
- A guard set $guard((q,q'))$ for each transition $(q,q') \in \mathcal{T}$;
- A jump function $jump_{(q,q')} : X \to \mathcal{P}(X)$ for each transition $(q,q') \in \mathcal{T}$;
- An output space $O \subseteq \mathbb{R}^l$;
- An output map $out : Q \times X \to O$.

For a given input function $u : \mathbb{R}^+ \to U$, which maps each point in time to an input value, a *state trace* x of H is

$$x = (q_0, x_0(.))(q_1, x_1(.))\ldots \tag{1}$$

with discrete states $q_i \in Q$, continuous state functions $x_i : [t_i, t_{i+1}] \to X$, and with the initial state $(q_0, x_0(0)) \in I_H$. The transitions from q_i to a new state q_{i+1} have to satisfy $(q_i, q_{i+1}) \in \mathcal{T}$. The continuous state function $x_i(.)$ has to satisfy the invariant set $x_i(t) \in$

$inv(q_i)$ and the differential inclusion $\dot{x}_i(t) \in F_H(q_i, x_i(t), u(t))$. Upon a discrete transition (q_i, q_{i+1}), the continuous state satisfies $x_i(t_{i+1}) \in guard((q_i, q_{i+1}))$ as well as $x_{i+1}(t_{i+1}) \in jump_{(q_i,q_{i+1})}(x_i(t_{i+1}))$. Note that we have a non-deterministic initial state, non-deterministic flow and jump functions, so there are multiple state traces possible for a given input trajectory $u(.)$. The set of state traces for a given hybrid automaton H with initial set I_H under input $u(.)$ is denoted by $straces(H, u(.), I_H)$.

While state traces represent the internal states of the system, they are not observable. Instead, we are able to observe the output trace $\tau : \mathbb{R}^+ \to O$ which is the mapping of the state trace x onto the observable output space via the map *out*:

$$\forall i \quad \forall t \in [t_i, t_{i+1}) : \qquad \tau(t) = out(q_i, x_i(t)). \tag{2}$$

The set of all output traces under an input trajectory $u(.)$ and the initial set I_H is denoted by $otraces(H, u(.), I_H)$. Therefore, the set $otraces(H, u(.), I_H)$ represents all possible observable behaviors over time for the given $u(.)$ and I_H. If $otraces(H, u(.), I_H)$ has one element only for every $u(.)$ and a single initial state, the hybrid automaton H is called *deterministic* and *non-deterministic* otherwise.

The already existing trace conformance, which we talked about in the overview section, can now be defined formally:

Definition 1 (trace conformance [4])**.** *Let S and M be two systems with the same input set and output space, and with the initial sets I_S and I_M, then S is trace conformant to M, which is denoted by $S \preceq_T M$, if*

$$otraces(S, u(.), I_S) \quad \subseteq \quad otraces(M, u(.), I_M) \tag{3}$$

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

This means when trace conformance holds, all observable behavior over time of S is also observable of M. A safety property consists of a set of unsafe output states B_t for every time t. If this set is never reachable, i.e.,

$$\forall \tau \in otraces(H, u(.), I_H) \quad \forall t \geq 0 : \qquad \tau(t) \notin B_t \,. \tag{4}$$

then H is considered safe.

Given such a safety property with B_t and a model M, verification deals with algorithmically checking that (4) holds. When reasoning about the future evolution of a system H, one has to consider all—potentially infinitely many—output traces. With infinitely many traces, dealing with the output traces directly is intractable. One important approach to solve this problem is to use reachability analysis. For one point in time t, the *reachable set* (or shorter: reachset) of outputs of the hybrid automaton H at time t contains all output states which are possible at time t for a given input trajectory $u(.)$:

$$Reach_t(H, u(.), I_H) \quad = \quad \{\tau(t) \mid \tau \in otraces(H, u(.), I_H)\}. \tag{5}$$

We call the sequence of these reachable sets $Reach_t(H, u(.), I_H)$ over time t as the *reach sequence* of H. Note that the elements of *otraces* are functions over time, whereas the set $Reach_t$ consists of output states for one point in time t only. Reachable sets can be used to reason about properties of H by verifying that no unsafe set B_t of a safety property is reachable:

$$\forall t \geq 0 : \qquad Reach_t(H, u(.), I_H) \cap B_t \quad = \quad \emptyset \,.$$

Since the reach sequence is an abstraction of the output traces $otraces(H, u(.), I_H)$, trace conformance is not the best relation for transference between reach sequences, as described in Section 4. Therefore, we present reachset conformance in the following section.

Let us now more formally and generally specify the problems addressed in this paper. Given a non-deterministic model M of a hybrid system S, the type of relation $S \preceq M$ between S and M needed to transfer any safety property ψ from M to S is:

$$S \preceq M \wedge M \models \psi \qquad \Rightarrow \qquad S \models \psi, \tag{6}$$

where $S \models \psi$ means system S has the property ψ.

Let us also introduce a Gaussian process (Section 6.4, [32]) with parameter vectors $P = (p_1, \dots, p_n)$ and $V = (v_1, \dots, v_n)$ as a function gp mapping the input p to a Gaussian distribution with mean $m(p)$ and variance $\sigma^2(p)$. The (p_1, v_1) are the samples, and the Gaussian process generalizes the mapping by estimating the similarity between different values for p. The functions $m(p)$ and $\sigma^2(p)$ are defined as [32]:

$$m(p) = \bar{k}(p)^T K^{-1} \bar{c}, \quad \sigma^2(p) = k(p,p) - \bar{k}(p)^T K^{-1} \bar{k}(p), \tag{7}$$

where ^{T} is the matrix transposition, and $k(p,p')$ is a kernel function defined (in our case) as

$$k(p,p') := \theta_0 \exp(-\theta_1 \|p - p'\|^2) + \theta_2 + \theta_3 p^T p', \tag{8}$$

$\bar{k}(p)^T = (k(p_1,p), \dots, k(p_s,p))$, $\bar{c} = (c_1, \dots, c_s)$, and the matrix K contains the entries $k(p_i,p_j)$ in the i^{th} row and j^{th} column. The parameters θ_i can be computed using hyperparameter optimization with P and V (Section 6.4.2, [32]).

4. Reachset Conformance

With the preliminaries from Section 3, we are now able to formally define the notion of reachset conformance.

Definition 2 (reachset conformance [3])**.** *Let S and M be two systems with the same input space and output space. Let I_S and I_M be the initial sets of S and M, respectively; then, S is reachset-conformant to M, denoted by $S \preceq_R M$, if for all possible inputs $u(.)$ and $t \geq 0$:*

$$Reach_t(S, u(.), I_S) \quad \subseteq \quad Reach_t(M, u(.), I_M). \tag{9}$$

Reachset conformance directly considers the non-determinism of models while still being able to transfer safety properties.

Proposition 1. *Let two systems S and M be given with $S \preceq_R M$ and initial sets I_S and I_M, respectively. Let a safety property with forbidden state sets B_t be given for all t. For any input trajectory $u(.)$, the following transference holds for every t:*

$$Reach_t(M, u(.), I_M) \cap B_t = \emptyset \quad \Rightarrow \quad Reach_t(S, u(.), I_S) \cap B_t = \emptyset. \tag{10}$$

Proof. Since S is reachset conformant to M and thus $Reach_t(S, u(.), I_S)$ is a subset of $Reach_t(M, u(.), I_M)$ for all t, the proposition follows immediately. □

The following theorem shows that reachset conformance is the natural choice for the transference of safety properties.

Theorem 1. *Let two systems S and M be given. The transference of safety properties is equivalent to reachset conformance:* (10) *holds for all t and all possible B_t $\Leftrightarrow$ $S \preceq_R M$.*

Proof. One direction follows from Proposition 1. Let us assume that (10) holds for all t and all possible B_t. Choosing $B'_t := \mathbb{R}^m \setminus Reach_t(M, u(.), I_M)$, which is the complement of the reachable set of M at time t, the intersection of B'_t and $Reach_t(M, u(.), I_M)$ is obviously empty. Since this property is transferable from M to S, the equation

$$Reach_t(S, u(.), I_S) \quad \cap \quad (\mathbb{R}^m \setminus Reach_t(M, u(.), I_M)) \quad = \quad \emptyset$$

holds, and thus, $Reach_t(S, u(.), I_S) \subseteq Reach_t(M, u(.), I_M)$. This works for every t, and thus, S is reachset conformant to M. □

Although we are mainly interested in the transference of safety properties, there are temporal fragments which transfer with reachset conformance. For instance, temporal properties formalizable in reachset temporal logic, which were introduced by Roehm et al. [33]. However, temporal properties cannot be transfered in general, as the reach sequence is an abstraction of the output traces. Reachset conformance is a weaker conformance notion than trace conformance:

Theorem 2. *Let S and M be two systems with the same input set and output space; then,*

$$S \preceq_T M \quad \Rightarrow \quad S \preceq_R M \tag{11}$$

holds. The converse holds if the system M (and thus S) is deterministic.

Proof. Let $u(.)$ be an input trajectory, t be a point in time, $y \in Reach_t(S, u(.), I_S)$ and $S \preceq_T M$. Then, there is a $\tau \in otraces(S, u(.), I_S)$ with $\tau(t) = y$. From $S \preceq_T M$, it follows that τ is also a trace of M and $y \in Reach_t(M, u(.), I_M)$. The proposition follows, because the aforementioned implication holds for all y, t, and $u(.)$. When the system M is deterministic, there is only one trace in $otraces(M, u(.), I_M)$, and the reachable set for any time consists of only one state. Hence, S has the same trace and is also deterministic. □

This shows that reachset conformance is weaker compared to trace conformance and that we can transfer properties between reachsets in cases where trace conformance does not hold.

5. Reachset Conformance Testing

In this section, we show how to check the reachset conformance of a real system S against a model M. Additionally, we introduce a robustness measure which quantifies conformance. Since proving a physical model against the real world is not possible, the goal is to check if the non-conformance $S \npreceq_R M$ can be shown by a counter-example for a given input $u(.)$. Hence, we have to prove that the negation of (9) holds, which is

$$\exists u(.) \in U(.) \quad \exists t \geq 0 : \qquad Reach_t(S, u(.), I_S) \quad \nsubseteq \quad Reach_t(M, u(.), I_M). \tag{12}$$

Our test to check reachset conformance consists of three steps:

1. Obtain measurements of the system S as an underapproximation $Reach_t^u(S, u(.), I_S) \subseteq Reach_t(S, u(.), I_S)$ of the reachable states of S for a finite set T of points in time $t \in T$.
2. Compute an overapproximation $Reach_t^o(M, u(.), I_M) \supseteq Reach_t(M, u(.), I_M)$ of the reachable set of M for all $t \in T$.
3. Check if $Reach_t^u(S, u(.), I_S) \nsubseteq Reach_t^o(M, u(.), I_M)$ holds for any $t \in T$.

If for any t a non-inclusion is found, a counter-example is found, and non-conformance is proven. In the following, we discuss the steps in detail.

5.1. Obtain Measurements of S

Real measurements are subject to noise, and we assume there exists an error ε, which bounds the deviation of all measurements (t_i, τ_i) to the true trace τ:

$$\max_i d_2(\tau_i - \tau(t_i)) \quad \leq \quad \varepsilon. \tag{13}$$

In our case, we consider $d_2(.)$ to be the Euclidean 2-norm. Taking all runs of S for the same input $u(.)$, this approach builds up reachable sets underapproximations.

5.2. Overapproximation of the Reachable Sets of M

An overapproximation of M can be efficiently computed using reachability analysis. Our work builds on the tool CORA [6] to compute reachable set overapproximations for hybrid automata with nonlinear continuous dynamics. CORA uses zonotopes to represent reachable sets due to their efficiency in linear transformations and Minkowski additions [34].

Definition 3 (Zonotope)**.** *An n-dimensional zonotope Z in generator representation (G-representation) is the set*

$$Z \quad = \quad z(c, \langle g_1, \dots, g_m \rangle) \quad := \quad \left\{ c + \sum_{i=1}^{m} \lambda_i g_i \middle| \lambda_i \in [-1, 1] \right\}, \tag{14}$$

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

5.3. Exclusion Check

For a given $t \in T$, we have to check if a given measurement τ_i with $t = t_i$ is excluded from $Reach^o_t(M, u(.), I_M)$. Since we consider the measurement error, we have to check that all possible candidates for the real value are not contained to prove exclusion. Hence, the ε ball around τ_i has to be completely outside the reachable set to prove that a counter-example exists.

The distance is important information, which we will use for the model adaptation. Therefore, we are using support functions to define a distance metric (note that the approach using support functions can be applied to other convex (reach-)sets representations as well) [35].

Definition 4 (robustness)**.** *Let a vector $d \in \mathbb{R}^n \setminus \{0\}$, a point $x \in \mathbb{R}^n$, and a zonotope Z with center c and m generators g_i be given. Then*

$$\rho_d(Z, x) \quad := \quad \frac{d^T c + \sum_{i=1}^{m} |d^T g_i| - d^T x}{\sqrt{d^T d}} \tag{15}$$

is the directed robustness of Z and x in direction d. The robustness of x and Z is defined as $\rho(Z, x) := \min_d \ \rho_d(Z, X)$.

If the robustness metric $\rho(Z, x)$ is negative, x lies outside of Z and the robustness gives the negative of the minimal distance between Z and x in the Euclidean norm. If x is contained in Z, the robustness is the distance of x to the surface of Z. Hence, the robustness metric enables us to check exclusion.

Theorem 3. *A point $x \in \mathbb{R}^n$ is not contained in a zonotope Z if and only if the robustness is negative:*

$$x \notin Z \quad \Leftrightarrow \quad \rho(Z, x) < 0 \quad \Leftrightarrow \quad \exists d : \rho_d(Z, x) < 0$$

Proof. Let us assume that $x \in Z$ and Z has the generators g_i. Then, there exists λ_i with $x = c + \sum \lambda_i g_i$ and $|\lambda_i| \leq 1$. For any d holds

$$\begin{aligned} \rho_d(Z, x) &= \frac{d^T c + \sum |d^T g_i| - d^T(c + \sum \lambda_i g_i)}{\sqrt{d^T d}} \\ &= \sum \frac{|d^T g_i| - \lambda_i \, d^T g_i}{\sqrt{d^T d}}) \geq 0 \end{aligned}$$

The other direction follows analogously. □

For a given point τ_i with error ε, we are able to show exclusion of the real measurement by checking $\rho(Z, \tau_i) > \varepsilon$ (see Proposition 5, [3]). Hence, $\rho(Z, \tau_i)$ has to be computed. This can be achieved by sampling directions d and approximating the robustness, as shown in [3]. Another approach is to use linear programming to find the direction d which minimizes $\rho_d(Z, x)$.

6. Model Adaptation

As the verification capabilities of a model are highly dependent on the sizes of the reachsets, the measure $m_{ver}(M)$ on the reachsets is used to determine the verification capabilities:

$$m_{ver}(M) \quad = \quad \operatorname*{avg}_{t} \mathrm{Vol}(Reach_t(M, u(.), I_M)), \tag{16}$$

where the volume function Vol is a metric on the reachable sets. Here, we are using the volume of the reachsets

$$\mathrm{Vol}(Reach_t(M, u(.), I_M)) = \int_{Reach_t(M,u(.),I_M)} 1dx, \tag{17}$$

but the P-radius [36] and F-radius [37] can be used as well in case of computational limitations. Similarly, the conformance measure $m_{conf}(M)$ is defined as

$$m_{conf}(M) \quad = \quad \min_{t_i, \tau_i} \rho(Reach_{t_i}(M, u(.), I_M), \tau_i) \tag{18}$$

to show how robust the model is conformant for given measurements (t_i, τ_i) of the real system S under input $u(.)$ and initial condition I_M. Hence, an optimal model M_g (with respect to Figure 1) can be defined as

$$M_g \quad := \quad \min_{M \in \mathcal{M}} m_{ver}(M), \text{ with } m_{conf}(M) \geq 0 \tag{19}$$

where $\mathcal{M}$ is the set of all possible models. For computational feasibility, we assume that $\mathcal{M}$ can be represented by a parametrization that is a surjective projection $\pi : \mathbb{R}^l \to \mathcal{M}, \pi(p) = M$. The idea is to represent possible amounts of non-determinism by parameters as shown in the following example.

Example 1. *Let us consider the toy example of a bouncing ball. At one point in time, it has a certain height h over the ground and a velocity v. Over time, it is accelerated by the gravity and bounces off when reaching the ground. As non-determinism can be involved in the acceleration (continuous part) and the bouncing off (discrete part), the possible choices of the non-determinism can be modeled with parameters $(p_1, p_2)^T \in \mathbb{R}^2$ resulting in the differential inclusions $\dot{h}(t) = v(t)$, $\dot{v}(t) \in [-8.5 - p_1, -8.5 + p_1]$ and jump function $jump(h, v) = (-h, [0.75 - p_2, 0.75 + p_2]v)$ with guard $h = 0$. Using measurements of a real bouncing ball, p_1 and p_2 can be obtained by solving* (19).

In this paper, Equation (19) is solved using Gaussian processes (see 3) and Bayesian optimization with inequality constraints [38]. The central idea of Bayesian optimization is to use existing function evaluations to build probabilistic regression models. These models are Gaussian processes and are used to select the next parameter to test. In our setting, the Gaussian processes gp_{conf} and gp_{ver} are built to approximate the conformance measure and the verification measure:

$$gp_{conf}(p) \approx \sqrt[3]{m_{conf}(\pi(p))} \quad \text{and} \quad gp_{ver}(p) \approx m_{ver}(\pi(p)).$$

As the most interesting region for gp_{conf} is near zero, approximating the cube root instead of $m_{conf}(\pi(p))$ has been shown beneficial in our application in Section 7 for the learning process.

The model adaptation works by executing the following steps:

1. Initialize vectors $P = p_1, \dots, p_n$ with random values and calculate the vectors $V = v_1, \dots, v_n, C = c_1, \dots, c_n$ via $v_i = m_{ver}(p_i), c_i = m_{conf}(p_i)$.
2. Generate gp_{conf} and gp_{ver} using P, V, and C.
3. Find p_{n+1} minimizing gp_{conf} with $gp_{ver} > 0$ using Bayesian optimization [38], add p_{n+1} to P, and add $v_{n+1} = m_{ver}(p_{n+1}), c_{n+1} = m_{conf}(p_{n+1})$ to V and C, respectively.

This iteration is completed iteratively until the probability is high that p_j with $v_j = \min_i v_i$ is the solution of (19). Using gp_{conf} and gp_{ver}, this is measured using

$$isMin(v_j) = 1 - \max_p P(gp_{ver}(p) < v_j)P(gp_{conf}(p) > 0)$$

as an end criterion.

7. Application of Reachset Conformance to an Autonomous Vehicle

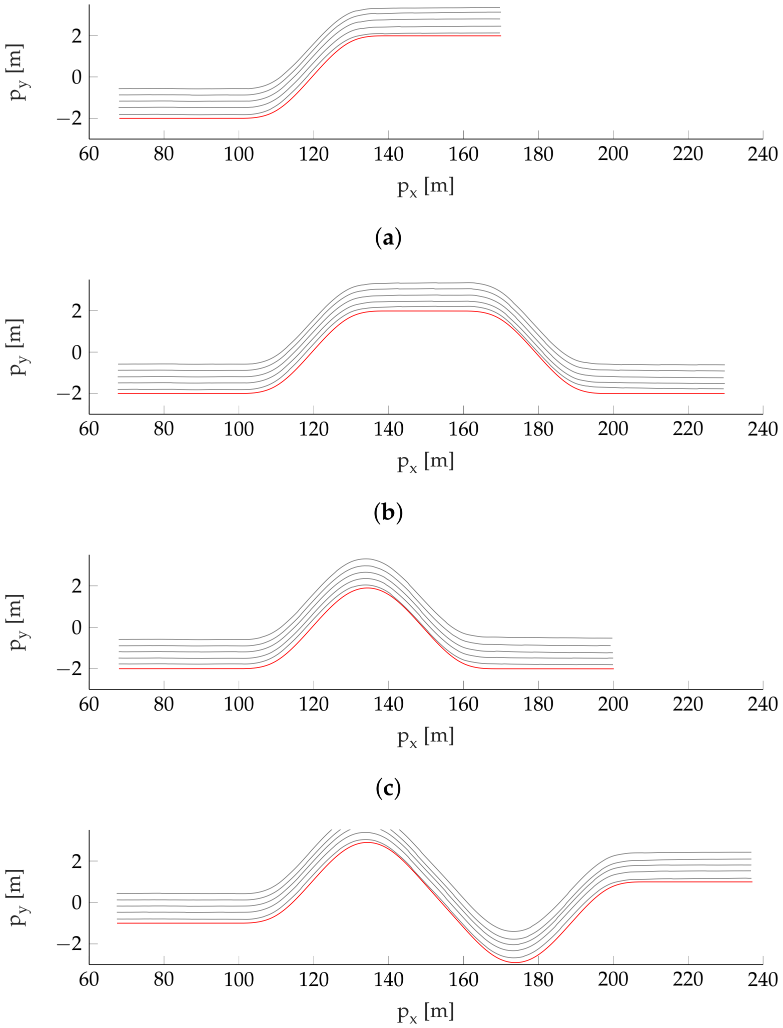
Automated vehicles are an important application of hybrid systems. One main verification task for automated vehicles is to ensure safe operation without collisions with other traffic participants. Since there are too many real-world situations to verify all of them beforehand, methods have been created to verify the automated vehicle online [39,40]. The verification approach is model-based, which creates the necessity to check that the model and the real vehicle are conformant such that verification results can be transfered.

The following demonstrates when to apply reachset conformance by measuring data of a real automated vehicle and building a reachset conformant model for it with the model adaptation method. As we have a limited amount of experimental data, one should increase the amount of measurements for real-world verification applications.

7.1. Experimental Setup

Four different types of maneuvers with a velocity of v_x = 10 m/s and a maximum lateral acceleration a_y = 2 m/s^2 have been considered. As visualized in Figure 3, the four maneuvers are

1. **Single lane-change maneuver:** One single lane-change from a right lane to the left lane, which is a typical maneuver for automated vehicles.
2. **Double lane-change maneuver:** After a single lane-change, the vehicle stays on the left lane for 4 s and switches back to the initial lane. This is a standard overtaking maneuver.
3. **Fast double lane-change maneuver:** This maneuver is similar to the double-lane change maneuver, but it immediately switches back to the right lane when on the left lane. Such a maneuver occurs when avoiding obstacles on the road and is more dynamic than the double-lane change.
4. **Slalom maneuver:** To challenge the model with measurements of a more dynamic maneuver, a slalom maneuver was additionally included.



(**a**)

(**b**)

(**c**)

(**d**)

Figure 3. The planned trajectory (red) and the driving data (gray, shifted by multiples of 30 cm in p_y for presentation purposes) for the maneuvers. (**a**) Single lane-change maneuver. (**b**) Double lane-change maneuver. (**c**) Fast double lane-change maneuver. (**d**) Slalom maneuver.

These maneuvers were selected based on the experimental capabilities of the driving location and can be seen as basic maneuvers in an urban multilane setting. Each maneuver was repeated five times with the average duration of a maneuver being 14.16 s at a rate of 100 Hz. Overall, the total driven distance of the dynamic maneuvers within the measurement data was around 3 km. The data were collected by Deutsches Zentrum für Luft- und Raumfahrt (DLR) with their test vehicle (FASCar II, a Volkswagen Passat TDI), which is equipped with a combined differential GPS receiver (DGPS) and inertial navigation system (INS). All maneuvers used for conformance testing have been executed in automated driving mode, i. e., closed-loop tracking of a predefined reference trajectory, which is sent from a PC to a closed-loop tracking controller on a dSPACE Autobox. We estimate the sensor error for the position of the vehicle by 5 cm and for the orientation of the vehicle by an angle of 0.5°.

7.2. Verification Model

The verification model contains of a steered vehicle model, which is combined with a tracking controller providing the steering inputs based on the ideal maneuver trajectory. Our vehicle model is based on the bicycle model [7,39]. The state space of the vehicle model is 6-dimensional and has the states $x = (p_x, p_y, \psi, v_x, v_y, \omega)^{\mathrm{T}}$, where p_x, p_y is the position of the vehicle's rear axle center in an earth-fixed coordinate system, and ψ is the orientation of the vehicle. The speed of the vehicle's rear axle center is given as $(v_x, v_y)^{\mathrm{T}}$ in the vehicle coordinate system. The velocity components are the respective projections to the vehicle's longitudinal and lateral axis. The vehicle's yaw rate is given as $\dot{\psi} = \omega$. The vehicle model's input vector $u = (u_a, u_\delta)^{\mathrm{T}}$ contains the longitudinal acceleration and the steering angle. The differential equations of the vehicle model are

$$\begin{aligned}
\dot{p}_x &= v_x \cos(\psi) - v_y \sin(\psi) + e_x \\
\dot{p}_y &= v_x \sin(\psi) + v_y \cos(\psi) + e_y \\
\dot{\psi} &= \omega + e_\psi \\
\dot{v}_x &= u_a + v_y \cdot \omega \\
\dot{v}_y &= f_f + f_r - v_x \cdot \omega - b \cdot \dot{\omega} \\
\dot{\omega} &= a \frac{m}{J} f_f - b \frac{m}{J} f_r \\
f_f &= -c_f \mu g \frac{b}{a+b} \left(\frac{v_y + (a+b) \cdot \omega}{v_x} - u_\delta \right) \\
f_r &= -c_r \mu g \frac{a}{a+b} \frac{v_y}{v_x}
\end{aligned}$$

with constants $J/m = 1.5$, $a = 1.16$, $b = 1.54$, $c_f = 10.8$, $c_r = 17.8$, $\mu = 0.8$, and $g = 9.81$.

The tracking controller by Hess et al. [41] was used consisting of a feed-forward controller and a PD feedback term for the deviation from the reference trajectory.

The state space of the combined model was divided into eight regions which represent the discrete states of the verification model. In each part, a Taylor expansion of the differential equations of the combined model is used as the differential equations of the verification model. Since the main dimensions of interest are the position p_x, p_y and orientation ψ of the vehicle, e.g., to detect possible collisions, these dimensions are used as the outputs and mapped onto this subspace with the output map out. The parameters e_x, e_y, and e_ω are injected as additive non-determinism $[-e_x, e_x]$, $[-e_y, e_y]$, and $[-e_\omega, e_\omega]$ into the differential equations for x, y, and ω, respectively.

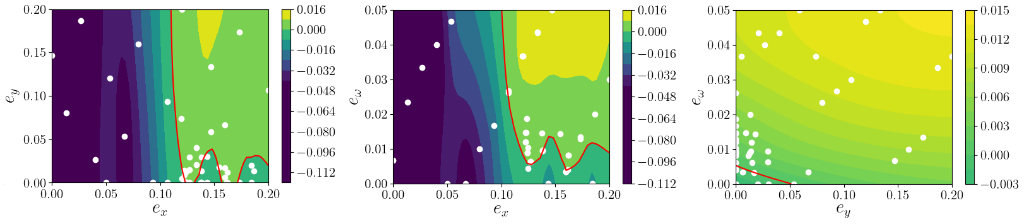
7.3. Reachset Conformance Testing

The initial points of all runs of a maneuver are used to build the initial set for the model for that maneuver. Since the measurements contain some sensor error, the bounding box of the initial points enlarged by the sensor errors is used as initial set I_M of the model. The pairwise direction check as described by Roehm et al. [3] is used to check for the exclusion of measured data from the reachable sets of the three-dimensional output space, considering the sensor error.

The model adaptation method from Section 6 has been applied to the model and the measurements of the automated vehicle. The measure gp_{conf} for the mapping from the parameters to the conformance measure after 30 iterations is visualized in Figure 4. In the figure, one can see the estimated robustness of each combination of non-deterministic bound parameters (e_x, e_y, e_ω). The red line consists of all parameters with $m_{conf}(M) = 0$, which are the boundary between the conformant and non-conformanct parameter areas. All parameter combinations on the upper right side of the red line can be considered as conformant.

The white points represent the parameter combinations with which the exclusion check has been executed to compute the robustness for the conformance measure. These combinations were iteratively selected by gp_{conf} and used to update gp_{conf}. As one can see in the figure, the white points are not uniformly distributed, but the density of points is much higher near certain areas of the red line. This is a direct outcome of the model adaptation algorithm. First, parameter combinations in the whole space of parameter combinations are selected to obtain an initial understanding of the regions of interest. After some iterations, the confidence of gp_{conf} increases, and more points near the expected parameter combinations of the optimal model M_g are selected. Please note that the white points in all subfigures are only projections of the three parameters to two parameters. Even when the white points are sitting on the non-conformant side of the red line in the subfigure, the non-projected parameters may not. In (a) and (b) in Figure 4, the red line is winding in the lower part. This is due to approximation errors in gp_{conf}. However, this is not a problem for the method, as these areas do not contain the optimal parameter combinations and thus are not explored further.

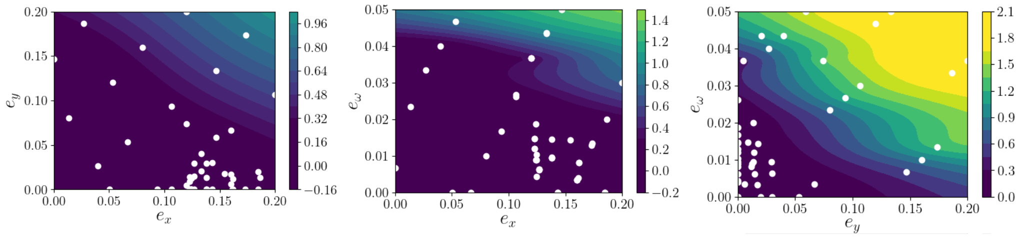
From Figure 4, the relation between the different parameters can be seen. The red line in the subfigures with e_x are mainly horizontal and vertical. This shows that the non-determinism on x is independed to the non-determinism on y and ψ. Contrary, e_y vs. e_ω shows that when the lateral nondeterminism is increased, the nondeterminism of the yaw rate can be reduced and vice versa. This shows that both have a similar impact on the lateral movement of the vehicle in our maneuvers and is likely a result of the main direction of travel in the x direction.



(**a**) (**b**) (**c**)

Figure 4. Conformance measure with respect to parameters as approximated by gp_{conf}. The red line is the boundary between conformant and non-conformant parameters. (**a**) e_x vs e_y, e_ω constant. (**b**) e_x vs e_ω, e_y constant. (**c**) e_y vs e_ω, e_x constant.

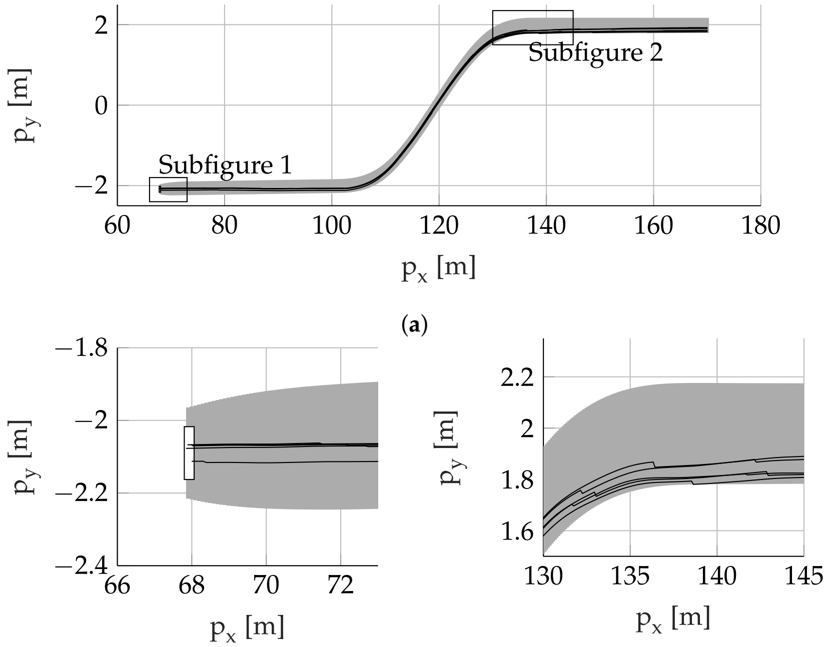
The verification measure with respect to the parameters is visualized in Figure 5. As in Figure 4, the verification measure is shown with the white points representing the parameter combinations used. The shape of all projections is looking quite similar. This is due to the monotonicity of the reachset sizes with respect to the non-determinism. When one parameter is increased, the overall non-determinism increases, and thus, so does the reachset size. In the origin, all parameters are zero, the model is deterministic, and the reach sequence reduces to a trace, cf. Theorem 2.



(**a**) (**b**) (**c**)

Figure 5. Verification measure with respect to parameters as approximated by gp_{ver}. (**a**) e_x vs e_y, e_ω constant. (**b**) e_x vs e_ω, e_y constant. (**c**) e_y vs e_ω, e_x constant.

The reachsets of the optimal model are visualized together with the measurement data in Figure 6. The optimal model has an interval width in the lateral position of 0.22 m and in the longitudinal position of 0.47 m, which can be considered as good enough for all considered driving tasks. Big uncertainty, such as over 0.5 m in the lateral position p_y, may lead to situations where the vehicle could potentially be already on an adjacent lane and a collision may be possible. Hence, we have built a reachset-conformant model which is amenable for verification purposes.



(**a**)

(**b**)

(**c**)

Figure 6. Projection of the measurements (black lines) and the reachsets of the model (gray area) to the position for the single lane-change maneuver. (**a**) Overview. (**b**) Subfigure 1: Zoom on the initial set (white box). (**c**) Subfigure 2: Zoom on the point in time, where the measured data comes closest to the reachset boundary.

8. Conclusions

In this paper, reachset conformance was presented that is able to relate a model to the system it models (and to other models as well). It was shown that reachset conformance is the natural conformance relation for safety properties, because safety properties transfer exactly in the case when reachset conformance does hold. Trace conformance implies reachset conformance and is the same in the case of deterministic systems.

Reachset conformance testing of a verification model is completed by searching for counter-examples with measurements of the real system. A robustness measure is introduced to estimate the distance of the model to be conformant or non-conformant based on the distance of a measurement to the reachable set of the verification model. Zonotopes are used as the representation of reachable sets which makes the computation of the robustness feasable.

A conformance measure is defined based on the robustness, and a verification measure is defined based on the size of the reachable sets and used to estimate the applicability of models for verification. The non-determinism of the model is considered as parametric, and a model adaptation algorithm is introduced to search for an optimal model, which minimizes the verification measure and has a positive value of the conformance measure. The algorithm uses Bayesian optimization to approximate the conformance measure and the verification measure and guides the search for the optimal parameters.

Finally, the presented methods are applied to an autonomous vehicle, for which data measurements of real driving maneuvers have been recorded. A parametric verification model is presented, and the methods of the paper are applied to find optimal parameters for that model to maximize the verification capabilities while ensuring conformance. Since the resulting reachable sets of the non-deterministic verification model have a size of at most 0.22 m in the lateral direction and 0.47 m in the longitudinal direction, the model produces small enough reachsets and can be used for verification purposes.

In future work, we want to collect extended amounts of driving data with a wider range of maneuvers and build a verified model that can be run online. This will combine multiple existing research directions and show how complicated it is to run the full pipeline for the verification of autonomous vehicles.

Author Contributions: Conceptualization, H.R.; methodology, H.R.; software, A.R. and H.R.; writing, H.R., A.R. and M.A.; visualization, H.R. and A.R.; supervision, M.A.; project administration, M.A.; funding acquisition, M.A.; All authors have read and agreed to the published version of the manuscript.

Funding: The authors gratefully acknowledge partial financial support by the project UnCoVerCPS funded by the European Commission under grant number 643921 and by the project KoSi funded by the German Federal Ministry for Digital and Transport under grant number O1MM20011A.

Data Availability Statement: Restrictions apply to the availability of the measured data. Data were obtained by Daniel Heß from Deutsches Zentrum für Luft- und Raumfahrt as part of the European Commission project UnCoVerCPS.

Acknowledgments: We thank Daniel Heß from DLR for providing the collected data. Also, we thank Hope el Saliente, Kira A Mi Lago, Canan Raisch, and Felix Werner for the helpful input and mental support.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Guéguen, H.; Lefebvre, M.A.; Zaytoon, J.; Nasri, O. Safety verification and reachability analysis for hybrid systems. *Annu. Rev. Control* **2009**, *33*, 25–36.
2. Frehse, G. An Introduction to Hybrid Automata, Numerical Simulation and Reachability Analysis. In *Proceedings of the Formal Modeling and Verification of Cyber-Physical Systems*; Drechsler, R., Kühne, U., Eds.; Springer: Berlin, Germany, 2015.
3. Roehm, H.; Oehlerking, J.; Woehrle, M.; Althoff, M. Reachset Conformance Testing of Hybrid Automata. In Proceedings of the HSCC, 2016; Vienna, Austria, April 12-14, 2016 pp. 277–286.
4. Dang, T. *Model-Based Testing for Embedded Systems*; CRC Press: Boca Raton, FL, USA, 2011; Chapter Model-based Testing of Hybrid Systems; pp. 383–423.
5. Schupp, S.; Ábrahám, E.; Chen, X.; Ben Makhlouf, I.; Frehse, G.; Sankaranarayanan, S.; Kowalewski, S. Current Challenges in the Verification of Hybrid Systems. In Proceedings of the Fifth Workshop on Design, Modeling and Evaluation of Cyber Physical Systems, Amsterdam, The Netherlands, 8 October 2015; pp. 8–24.
6. Althoff, M. An Introduction to CORA 2015. In Proceedings of the Workshop on Applied Verification for Continuous and Hybrid Systems, Brussels, Belgium, 9 July 2015; pp. 120–151.
7. Althoff, M.; Dolan, J.M. Reachability Computation of Low-Order Models for the Safety Verification of High-Order Road Vehicle Models. In Proceedings of the American Control Conference, Atlanta, GA, USA, 8–10 June 2012; pp. 3559–3566.
8. Roehm, H.; Oehlerking, J.; Woehrle, M.; Althoff, M. Model Conformance for Cyber-Physical Systems: A Survey. *ACM Trans. Cyber Phys. Syst.* **2019**, *3*, 30:1–30:26.
9. van Osch, M.P.W.J. Automated Model-based Testing of Hybrid Systems. Ph.D. Thesis, Technische Universiteit Eindhoven: Eindhoven, The Netherlands, 2009.
10. Tretmans, G.J. A Formal Approach to Conformance Testing. Ph.D Thesis, Universiteit Twente, Twente, The Netherlands, 1992.
11. Abbas, H.; Mittelmann, H.; Fainekos, G. Formal Property Verification in a Conformance Testing Framework. In Proceedings of the 12th ACM/IEEE International Conference on Formal Methods and Models for Codesign, Lausanne, Switzerland, 19–21 October 2014; pp. 155–164.
12. Abbas, H.; Hoxha, B.; Fainekos, G.; Deshmukh, J.V.; Kapinski, J.; Ueda, K. Conformance Testing as Falsification for Cyber-Physical Systems. *arXiv* **2014**, arXiv:1401.5200.
13. Annapureddy, Y.S.R.; Fainekos, G.E. Ant Colonies for Temporal Logic Falsification of Hybrid Systems. In Proceedings of the 36th Annual Conference of IEEE Industrial Electronics, Glendale, AZ, USA, 7–10 November 2010; pp. 91–96.
14. Quesel, J.D. Similarity, Logic, and Games: Bridging Modeling Layers of Hybrid Systems. Ph.D. Thesis, University of Oldenburg, 2013. Carl-von-Ossietzky-Universität Oldenburg; Nr. [20]12,03
15. Deshmukh, J.V.; Majumdar, R.; Prabhu, V.S. Quantifying Conformance Using the Skorokhod Metric. In Proceedings of the CAV, 2015, San Francisco, CA, USA, July 18-24, 2015.
16. Majumdar, R.; Prabhu, V.S. Computing Distances between Reach Flowpipes. In Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control. Association for Computing Machinery, Vienna, Austria, 12–14 April 2016; pp. 267–276. https://doi.org/10.1145/2883817.2883850.
17. Frehse, G. Compositional Verification of Hybrid Systems using Simulation Relations. Ph.D. Thesis, Radboud Universiteit Nijmegen, Nijmegen, The Netherlands, 2005.

18. Tabuada, P. *Verification and Control of Hybrid Systems—A Symbolic Approach*; Springer: Berlin, Germany, 2009.
19. van der Schaft, A. Bisimulation of Dynamical Systems. In Proceedings of the Hybrid Systems: Computation and Control, Philadelphia, PA, USA, 25–27 March 2004; pp. 555–569.
20. Bujorianu, M.L.; Lygeros, J.; Bujorianu, M.C. Bisimulation for general stochastic hybrid systems. In Proceedings of the HSCC, 2005, Zurich, Switzerland, March 9-11, 2005.
21. Cuijpers, P.J.L. On Bicontinuous Bisimulation and the Preservation of Stability. In Proceedings of the Hybrid Systems: Computation and Control, Pisa, Italy, 3–5 April 2007; pp. 676–679.
22. Prabhakar, P.; Dullerud, G.; Viswanathan, M. Stability Preserving Simulations and Bisimulations for Hybrid Systems. *IEEE Trans. Autom. Control* **2015**, *60*, 3210–3225. https://doi.org/10.1109/TAC.2015.2422431.
23. Girard, A.; Julius, A.A.; Pappas, G.J. Approximate simulation relations for hybrid systems. *IFAC Proc. Vol.* **2006**, *39*, 106–111.
24. Girard, A.; Julius, A.A.; Pappas, G.J. Approximate simulation relations for hybrid systems. *Discret. Event Dyn. Syst.* **2008**, *18*, 163–179.
25. Tabuada, P. Approximate simulation relations and finite abstractions of quantized control systems. In Proceedings of the HSCC, 2007, Pisa, Italy, April 3-5, 2007.
26. Liu, S.B.; Althoff, M. Reachset Conformance of Forward Dynamic Models for the Formal Analysis of Robots. In Proceedings of the P IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018; pp. 370–376.
27. Kochdumper, N.; Tarraf, A.; Rechmal, M.; Olbrich, M.; Hedrich, L.; Althoff, M. Establishing Reachset Conformance for the Formal Analysis of Analog Circuits. In Proceedings of the 25th Asia and South Pacific Design Automation Conference, Beijing, China, 13–16 January 2020; pp. 199–204.
28. Bravo, J.M.; Alamo, T.; Camacho, E.F. Bounded Error Identification of Systems With Time-Varying Parameters. *IEEE Trans. Autom. Control* **2006**, *51*, 1144–1150.
29. Wang, H.; Kolmanovsky, I.V.; Sun, J. Zonotope-based recursive estimation of the feasible solution set for linear static systems with additive and multiplicative uncertainties. *Automatica* **2018**, *95*, 236–245.
30. Liu, B.; Kong, S.; Gao, S.; Zuliani, P.; Clarke, E.M. Parameter Synthesis for Cardiac Cell Hybrid Models Using d-Decisions. In *International Conference on Computational Methods in Systems Biology;* Springer: Cham, Switzerland,2014.
31. Alur, R.; Coucoubetis, C.; Halbwachs, N.; Henzinger, T.A.; Ho, P.H.; Nicolin, X.; Olivero, A.; Sifakis, J.; Yovine, S. The Algorithmic Analysis of Hybrid Systems. *Theor. Comput. Sci.* **1995**, *138*, 3–34.
32. Bishop, C. *Pattern Recognition and Machine Learning*; Information Science and Statistics; Springer: Berlin, Germany, 2006.
33. Roehm, H.; Oehlerking, J.; Heinz, T.; Althoff, M. STL Model Checking of Continuous and Hybrid Systems. In Proceedings of the ATVA, 2016; Volume 9938, pp. 412–427.
34. Althoff, M.; Stursberg, O.; Buss, M. Computing Reachable Sets of Hybrid Systems Using a Combination of Zonotopes and Polytopes. *Nonlinear Anal. Hybrid Syst.* **2010**, *4*, 233–249.
35. Girard, A.; Le Guernic, C.; Maler, O. Efficient Computation of Reachable Sets of Linear Time-Invariant Systems with Inputs. In *Hybrid Systems: Computation and Control*; LNCS 3927; Springer: Berlin, Germany, 2006; pp. 257–271.
36. Le, V.T.H.; Stoica, C.; Alamo, T.; Camacho, E.F.; Dumur, D. Zonotopic Guaranteed State Estimation for Uncertain Systems. *Automatica* **2013**, *49*, 3418–3424.
37. Alamo, T.; Bravo, J.M.; Camacho, E.F. Guaranteed State Estimation by Zonotopes, In Proceedings of the 42nd IEEE International Conference on Decision and Control, 2003, Maui, HI, USA, 09-12 December 2003. pp. 5831–5836.
38. Gardner, J.R.; Kusner, M.J.; Xu, Z.; Weinberger, K.Q.; Cunningham, J.P. Bayesian Optimization with Inequality Constraints. In Proceedings of the 31st I nternational Conference on International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. II-937–II-945.
39. Althoff, M.; Dolan, J.M. Set-Based Computation of Vehicle Behaviors for the Online Verification of Autonomous Vehicles. In Proceedings of the 14th IEEE Conference on Intelligent Transportation Systems, Washington, DC, USA, 5–7 October 2011; pp. 1162–1167.
40. Althoff, M.; Dolan, J.M. Online Verification of Automated Road Vehicles Using Reachability Analysis. *IEEE Trans. Robot.* **2014**, *30*, 903–918.
41. Heß, D.; Löper, C.; Hesse, T. Safe Cooperation of Automated Vehicles. In Proceedings of the AAET—-Automatisiertes und vernetztes Fahren, Braunschweig, Germany, 8–9 February 2017.