

Reachset-Conformant System Identification

Laura Lützow and Matthias Althoff

Abstract—Formal verification techniques play a pivotal role in ensuring the safety of complex cyber-physical systems. To transfer model-based verification results to the real world, we require that the measurements of the target system lie in the set of reachable outputs of the corresponding model, a property we refer to as reachset conformance. This paper is on automatically identifying those reachset-conformant models. While state-of-the-art reachset-conformant identification methods focus on linear state-space models, we generalize these methods to non-linear state-space models and linear and nonlinear input-output models. Furthermore, our identification framework adapts to different levels of prior knowledge on the system dynamics. In particular, we identify the set of model uncertainties for white-box models, the parameters and the set of model uncertainties for gray-box models, and entire reachset-conformant black-box models from data. The robustness and efficacy of our framework are demonstrated in extensive numerical experiments using simulated and real-world data.

Index Terms—Data-defined modeling, dynamical systems, formal safety verification, input-output models, reachability analysis, reachset conformance, set-based computing, uncertainty quantification, zonotopes.

I. INTRODUCTION

Formal verification techniques require mathematical models that describe the behavior of the target system [1]. Since no model of a cyber-physical system is completely accurate due to the inherent complexity of real-world dynamics, we include uncertainties in our models to cover the discrepancies to the target system. The automatic identification of these uncertainties alongside the nominal model is the topic of this article.

A. Literature Overview

Classical system identification techniques aim to find the model or model parameters, which minimize some objective function, usually by making probabilistic assumptions about the uncertainty in the data and the model [2]–[5]. In many applications, however, we do not have any information about the underlying probability distributions [6]. Furthermore, to establish rigorous safety guarantees in planning and control, it is beneficial to have models with bounded uncertainties [7].

This paper was submitted for review on ... This work was funded in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1608 – under grant number 501798263 and in part by the European Commission project justITSELF under grant number 817629.

The authors are both with the School of Computation, Information and Technology, Technical University of Munich, 85748 Garching, Germany. {laura.luetzow@tum.de, althoff@in.tum.de}

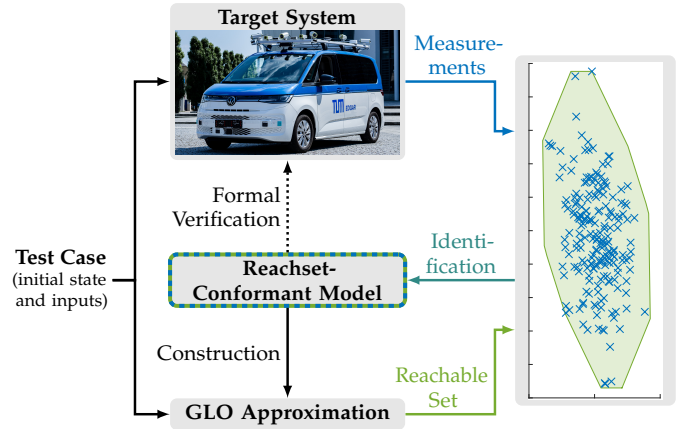


Fig. 1. Identification of a reachset-conformant model: The outputs of the target system are measured across different test cases, each defined by an initial state and an input trajectory. A reachset-conformant model is identified by adjusting its parameters and uncertainty sets such that all measurements lie within the model's reachable set, which is computed for each test case using the corresponding GLO approximation.

Given the model structure and the bounds of an additive model uncertainty, which describes the errors between the model outputs and the measurements of the target system, the set of feasible parameters can be estimated by *set-membership methods*, also called bounded-error approaches [8]. This set of feasible parameters is overapproximated by multi-dimensional intervals [9]–[11], ellipsoids [6], or zonotopes [12]. An overapproximation with arbitrary accuracy can be obtained using unions of non-overlapping multi-dimensional intervals [13]–[15]. For linear systems, the exact parameter set can be represented by a polytope [16] if the model uncertainty is bounded by an interval, or by a matrix zonotope [17] in the case of additive zonotopic uncertainties. However, specifying the model structure and the uncertainty bounds a priori, as required for set-membership approaches, is usually a difficult task for real-world systems. If the uncertainty bounds are assumed too large, the resulting set of possible models is also too large. On the other hand, we might not find any feasible parameters if the uncertainty bounds are assumed too small.

To address these problems, techniques for *conformant model identification* have been developed [1]. These methods find optimal models and their corresponding uncertainties such that a conformance relation is established. Depending on the conformance relation, different properties can be transferred from the conformant model to the system. Many publications focus on *simulation relations*, which facilitates the transfer of temporal logic properties by relating system states [18]–[20]. Establishing a simulation relation, however, requires that

the system state is observable. Less restrictive are trace conformance and reachset conformance, which only consider the output of the system. In *trace-conformant* identification, the model and a bounded set for the uncertainties, which we refer to as uncertainty set, are constructed such that the model is able to produce all output traces of the target system [21]–[23]. In most cases, this can only be done approximately, which strongly limits the usability of the resulting models for formal verification. *Reachset conformance*, on the other hand, only requires that the system output is contained in the reachable set of the model at each time step, which facilitates using simpler model structures and smaller uncertainties. As reachset conformance transfers safety properties [24], a reachset-conformant model suffices for formal safety verification, e.g., for collision avoidance [25] or the computation of control invariant sets [26].

Although reachset-conformant identification is a young research direction, there already exist multiple approaches to identify reachset-conformant state-space models in the literature: A given model can be made reachset-conformant by incrementally enlarging additive uncertainty sets until all measurements are contained in the reachable set [27]. If we have estimates for the system state, we can separately compute the additive process and measurement uncertainty set as multi-dimensional intervals that contain all errors between the system states and the states predicted by the model, and between the measurements and the model outputs [28]. Given full state measurements, the reachset-conformant identification of linear models with additive, zonotopic uncertainty sets can be formulated as a convex programming problem minimizing the size of the uncertainty sets [29]. Furthermore, additive, zonotopic uncertainty sets for linear systems can be computed with a linear program that minimizes the interval norm of the resulting reachable set [30]. This method can be extended to the identification of the initial state uncertainty set and the minimization of the Frobenius norm of the reachable set [31]. The scalability of the linear program with respect to the system dimension and time horizon can be improved by using the generator representation instead of the halfspace representation of the reachable set [32], while recursive identification techniques improve the scalability with respect to the size of the identification dataset [33].

B. Contributions and Outline

To the best of our knowledge, there exist no publications about a) the reachset-conformant identification of arbitrary (non-additive) uncertainty sets, b) the reachset-conformant identification of input-output models such as autoregressive models with exogenous inputs (ARX models) or nonlinear autoregressive model with exogenous inputs (NARX models), and c) reachset-conformant black-box identification based entirely on data. Our work addresses these topics by proposing a general framework for reachset-conformant system identification, as visualized in Fig. 1. In particular, our contributions are as follows:

- We introduce a general linear output (GLO) approximation and derive its construction for state-space and input-output models.

- We compute, for the first time, the reachable set of nonlinear input-output models using the GLO approximation.
- Furthermore, we present a linear program for the reachset-conformant white-box identification of linear and nonlinear models, i.e., we identify uncertainty sets such that reachset conformance is established. To the best of our knowledge, this is the first approach to estimate, potentially non-additive, uncertainty sets for input-output models and nonlinear state-space models.
- Additionally, we propose a framework for the identification of reachset-conformant models from data with little (gray-box identification) or no prior knowledge about the system dynamics (black-box identification).
- The performance and scalability of the proposed approaches are analyzed in numerical experiments for different dynamical systems and considering different levels of prior knowledge.
- We demonstrate real-world applicability by identifying a reachset-conformant vehicle model from real-world data.

This article is structured as follows: Fundamentals about zonotopes and reachset conformance and the problem statement are introduced in Sec. II. In Sec. III, we present a general approach to compute the reachable set of different model types using the novel GLO approximation. Sec. IV deals with the reachset-conformant identification of uncertainty sets for given model dynamics. Subsequently, we extend this approach to cases where the model dynamics contain unknown parameters in Sec. V and cases where we do not have any knowledge about the model in Sec. VI. The efficacy of our approaches is demonstrated in numerical experiments in Sec. VII. Sec. VIII concludes this article.

C. Notation

Sets are denoted by calligraphic letters, matrices by uppercase letters, and vectors and scalars by lowercase letters. Furthermore, we use $\mathbf{1}$, $\mathbf{I}$, and $\mathbf{0}$, respectively, for a vector filled with ones, the identity matrix, and a matrix of zeros, where the matrix dimensions can be explicitly written in the subscript. The operation $\text{diag}(v)$ returns a diagonal matrix with the elements of the vector v on its main diagonal and $\text{diag}(A_1, A_2, \dots, A_n)$ represents a blockdiagonal matrix where the matrices $A_1, A_2, \dots, A_n$ are concatenated diagonally. Additionally, we introduce the notation $v_{0:k}$ for the vertical concatenation $[v_0^\top v_1^\top \dots v_k^\top]^\top$ of the time-varying vector $v_i \in \mathbb{R}^m$ for the time steps $i = 0, \dots, k$. The Jacobian of $f(v_{0:k}): \mathbb{R}^{(k+1)m} \rightarrow \mathbb{R}^n$ with respect to the vector v_i is written as $\nabla_{v_i} f = [\frac{\partial f}{\partial v_{i,1}} \dots \frac{\partial f}{\partial v_{i,m}}]$, where $v_{i,j}$ denotes the j -th element of v_i .

Let us also introduce the Minkowski sum of two sets $S_a, S_b \subset \mathbb{R}^n$ as $S_a \oplus S_b = \{s_a + s_b | s_a \in S_a, s_b \in S_b\}$, the Cartesian product $S_a \times S_b = \{[s_a^\top s_b^\top]^\top | s_a \in S_a, s_b \in S_b\}$, and the linear transformation of S_a using the matrix $A \in \mathbb{R}^{m \times n}$ as $AS_a = \{As | s \in S_a\}$. The Cartesian product $S_0 \times S_1 \times \dots \times S_k$ of the sequence of sets $S_i, i = 0, \dots, k$, is denoted by $S_{0:k}$.

II. PRELIMINARIES AND PROBLEM STATEMENT

We first introduce zonotopes, which is the set representation used in this work, followed by the definition of reachset conformance. Subsequently, we formulate our problem statement.

A. Zonotopes

A zonotope is a convex set representation that is closed under Minkowski sum and linear map. The generator representation of zonotopes makes it possible to efficiently compute the aforementioned operations in high dimensions [34]:

Definition 1 (Generator Representation of Zonotopes [35]). *A zonotope $\mathcal{Z} \subset \mathbb{R}^n$ can be described by its center vector $c = \text{cen}(\mathcal{Z}) \in \mathbb{R}^n$ and its generator matrix $G = \text{gen}(\mathcal{Z}) = [g_1 \ \dots \ g_\eta] \in \mathbb{R}^{n \times \eta}$ as*

$$\mathcal{Z} = \left\{ c + \sum_{i=1}^{\eta} \lambda_i g_i \mid \lambda_i \in [-1, 1] \right\} = \langle c, G \rangle.$$

Alternatively, zonotopes can be described by their halfspace representation:

Definition 2 (Halfspace Representation of Zonotopes [36]). *A zonotope $\mathcal{Z} \subset \mathbb{R}^n$ can be represented as the intersection of a finite number of halfspaces, whose normal vectors are concatenated row-wise in the matrix $N \in \mathbb{R}^{h \times n}$ and whose offsets to the origin are stacked in the vector $d \in \mathbb{R}^h$:*

$$\mathcal{Z} = \{x \in \mathbb{R}^n \mid Nx \leq d\}.$$

A zonotope in generator representation can be converted to its halfspace representation using [36, Thm. 7]. Generally, computations using the generator representation scale better with increasing dimension than using the halfspace representation, since a zonotope with η generators has up to $2\binom{\eta}{n-1}$ halfspaces [36] and thus, the number of halfspaces h grows exponentially with the dimension n . Furthermore, the set operations Minkowski sum, Cartesian product, and linear transformation can be efficiently performed in generator representation: For two zonotopes $\mathcal{Z}_a = \langle c_a, G_a \rangle$ and $\mathcal{Z}_b = \langle c_b, G_b \rangle$ and the matrix A , we have $\mathcal{Z}_a \oplus \mathcal{Z}_b = \langle c_a + c_b, [G_a \ G_b] \rangle$, $\mathcal{Z}_a \times \mathcal{Z}_b = \langle [c_a^\top \ c_b^\top]^\top, \text{diag}(G_a, G_b) \rangle$, and $A\mathcal{Z}_a = \langle Ac_a, AG_a \rangle$. The size of a zonotope can be evaluated using the interval norm:

Definition 3 (Interval Norm of Zonotopes [31, Sec. 3.1]). *The interval norm for the zonotope $\mathcal{Z} = \langle c, G \rangle$ is defined as*

$$\|\mathcal{Z}\|_I = \mathbf{1}^\top |G| \mathbf{1}.$$

B. Reachset Conformance

As measurements are usually obtained in discrete time, we only consider discrete-time systems. Let us now introduce the output $\tau_k(S, x_0, u_{0:k})$ at time step k of a system S starting at the state x_0 , controlled by the inputs stacked in $u_{0:k} = [u_0^\top \ \dots \ u_k^\top]^\top$. The input vector u_i consists of the control signals and disturbances at time step i . We consider uncertainties by assuming that the initial state x_0 and inputs u_i , $i = 0, \dots, k$, deviate from the expected initial state $x_{*,0}$ and inputs $u_{*,i}$, which we call the nominal initial state and

the nominal inputs. These deviations are represented by the sets $\mathcal{X}_0(S)$ and $\mathcal{U}_i(S)$. The nominal initial state and inputs are collected in a test case:

Definition 4 (Test Case). *A test case labeled by $m \in \mathcal{M}$ of length n_k consists of the nominal initial state $x_{*,0}^{(m)}$ and the nominal inputs $u_{*,0:n_k-1}^{(m)}$.*

Each test case is executed n_s times and the resulting outputs are measured as $y_k^{(m,s)}(S)$, with $k = 0, \dots, n_k - 1$, $s = 1, \dots, n_s$. While the number of executions n_s can vary for different test cases m , we assume it to be constant in this paper for notational simplicity.

We refer to the set of reachable outputs as the reachable set, which can be formally defined as follows:

Definition 5 (Reachable Set). *The reachable set of the system S for test case m and at time step k is defined as*

$$\mathcal{Y}_k^{(m)}(S) := \{ \tau_k(S, x_0, u_{0:k}) \mid x_0 \in x_{*,0}^{(m)} \oplus \mathcal{X}_0^{(m)}(S), \\ u_{0:k} \in u_{*,0:k}^{(m)} \oplus \mathcal{U}_{0:k}^{(m)}(S) \}.$$

Subsequently, we consider a real-world system or a high-fidelity model as the target system S_T . For many applications, such as real-time verification, we need a simple model S_M that is able to approximate the behavior of S_T . To transfer safety properties from the model S_M to the target system S_T , reachset conformance is necessary and sufficient [24, Thm. 1]:

Definition 6 (Reachset Conformance [1, Sec. 3.5]). *System S_M is a reachset-conformant model of the target system S_T if and only if for all time steps k and all possible test cases m , the reachable set $\mathcal{Y}_k^{(m)}(S_M)$ contains the reachable set $\mathcal{Y}_k^{(m)}(S_T)$ of S_T :*

$$S_T \text{ conf}_R S_M \Leftrightarrow \forall m, k: \mathcal{Y}_k^{(m)}(S_T) \subseteq \mathcal{Y}_k^{(m)}(S_M). \quad (1)$$

Since the reachable set does not exist for real-world systems and cannot be computed for many models [37], [38], we use the following approximations: The reachable set of the target system S_T will be approximated by measurements $y_k^{(m,s)}(S_T)$, i.e.,

$$\mathcal{Y}_k^{(m)}(S_T) = \lim_{n_s \rightarrow \infty} \{ y_k^{(m,s)}(S_T) = \tau_k(S_T, x_0^{(m,s)}, u_{0:k}^{(m,s)}) \mid \\ s = 1, \dots, n_s, x_0^{(m,s)} \in x_{*,0}^{(m)} \oplus \mathcal{X}_0^{(m)}(S_T), \\ u_{0:k}^{(m,s)} \in u_{*,0:k}^{(m)} \oplus \mathcal{U}_{0:k}^{(m)}(S_T) \}. \quad (2)$$

As we can only consider a finite number of samples n_s in practice, reachset conformance can only be checked for the given samples but not be proven in general. Furthermore, we might use the reachable set $\hat{\mathcal{Y}}_k^{(m)}(S_M)$ computed by an over-approximative reachability algorithm instead of $\mathcal{Y}_k^{(m)}(S_M)$ in (1). Safety properties can be transferred if the reachability algorithm used for verification is identical or even more over-approximative than the algorithm that was used for computing $\hat{\mathcal{Y}}_k^{(m)}(S_M)$ when establishing reachset conformance.

C. Problem Statement

We present methods for an efficient identification of reachset-conformant models from data. A model S_M is characterized by a set of model functions $\mathcal{F}$, which describe the model dynamics and make it possible to predict the output $\tau_k(S_M, x_0, u_{0:k})$ (see Sec. III for definitions for different model types), and its uncertainty sets $\mathcal{X}_0^{(m)}(S_M)$ and $\mathcal{U}_i^{(m)}(S_M)$. We assume that the uncertainty sets can be represented by test case- and time step-independent zonotopes. The center vectors consist of the initial estimates $c_x \in \mathbb{R}^{n_x}$ and $c_u \in \mathbb{R}^{n_u}$ and unknown center shifts $c_{\Delta,x} \in \mathbb{R}^{n_x}$ and $c_{\Delta,u} \in \mathbb{R}^{n_u}$. The generator matrices are constructed from known templates $G_x \in \mathbb{R}^{n_x \times \eta_x}$ and $G_u \in \mathbb{R}^{n_u \times \eta_u}$, scaled by unknown factors $\alpha_x \in \mathbb{R}_{\geq 0}^{\eta_x}$ and $\alpha_u \in \mathbb{R}_{\geq 0}^{\eta_u}$. This results in the following uncertainty sets:

$$\mathcal{X}_0^{(m)}(S_M) = \langle c_x + c_{\Delta,x}, G_x \text{diag}(\alpha_x) \rangle, \quad (3a)$$

$$\mathcal{U}_i^{(m)}(S_M) = \langle c_u + c_{\Delta,u}, G_u \text{diag}(\alpha_u) \rangle, \quad (3b)$$

for all time steps $k = 0, \dots, n_k - 1$ and test cases m . For brevity, we introduce the combined variables $c = [c_x^\top \ c_u^\top]^\top$, $c_{\Delta} = [c_{\Delta,x}^\top \ c_{\Delta,u}^\top]^\top$, and $\alpha = [\alpha_x^\top \ \alpha_u^\top]^\top$.

We can now formulate our problem statement of identifying a reachset-conformant model. Since unnecessary conservatism impedes the use of reachability analysis, we want the model to produce tight reachable sets. Thus, we additionally minimize a cost function that evaluates the size of the reachable sets for all test cases.

Problem 1 (Reachset-conformant Identification).

We establish reachset conformance between the model S_M with the uncertainty sets in (3) and the target system S_T for the test cases in $\mathcal{M}$. This is achieved by solving the optimization problem

$$q^* = \arg \min_q \sum_{m \in \mathcal{M}} \sum_{k=0}^{n_k-1} w_k r(\hat{\mathcal{Y}}_k^{(m)}(S_M)) \quad (4a)$$

$$\text{s.t.} \quad \forall m, k, s: y_k^{(m,s)}(S_T) \in \hat{\mathcal{Y}}_k^{(m)}(S_M), \quad (4b)$$

where q are unknown parameters of the uncertainty sets and the model functions of S_M , the function $r(\mathcal{Y})$ evaluates the size of a set $\mathcal{Y}$, and $w_k \geq 0$ is a weight for time step k .

From this general problem, we derive the following sub-problems considered in this work, which depend on the variables we optimize:

Problem 1.1 (Identification of White-Box Models).

We solve Problem 1 with the optimization variables $q = (\alpha, c_{\Delta})$, assuming the model functions $\mathcal{F}$ are known.

Problem 1.2 (Identification of Gray-Box Models).

We solve Problem 1 with the optimization variables $q = (\alpha, c_{\Delta}, p)$, where the model functions $\mathcal{F} = \mathcal{F}_p$ depend on unknown model parameters p .

Problem 1.3 (Identification of Black-Box Models).

We solve Problem 1 with the optimization variables $q = (\alpha, c_{\Delta}, \mathcal{F})$.

Estimating arbitrary model parameters p in Problem 1.2 or the functions $\mathcal{F}$ in Problem 1.3 are complex, nonlinear programming problems. As computing the global optimal solution of general nonlinear programming problems is NP-hard and hence intractable, we will focus on efficiently finding a good local optimum that satisfies the conformance constraints.

Note that we will omit the notation (S_M) for variables referring to the model S_M in the remainder of this work to enhance readability.

III. REACHABILITY ANALYSIS

In this section, we explain how to generalize reachability analysis for different model types using a so-called general linear output approximation.

Definition 7 (General Linear Output (GLO) Approximation). The GLO approximation

$$\begin{aligned} \bar{\tau}_k^{(m)}(x_0, u_{0:k}) &= \bar{y}_k^{(m)} + \bar{C}_k^{(m)}(x_0 - \bar{x}_0^{(m)}) \\ &\quad + \sum_{i=0}^k \bar{D}_{k,i}^{(m)}(u_i - \bar{u}_i^{(m)}) \end{aligned} \quad (5)$$

is defined by the reference output $\bar{y}_k^{(m)}$, the system matrix $\bar{C}_k^{(m)}$, and the input matrices $\bar{D}_{k,i}^{(m)}$. It provides a linear approximation of the output τ_k along the reference trajectory defined by $\bar{x}_0^{(m)}$ and $\bar{u}_{0:k}^{(m)}$. The deviation between the output τ_k and the linearized output $\bar{\tau}_k^{(m)}$ is described by the linearization error $\bar{e}_k^{(m)}$:

$$\bar{e}_k^{(m)}(x_0, u_{0:k}) = \tau_k(x_0, u_{0:k}) - \bar{\tau}_k^{(m)}(x_0, u_{0:k}).$$

Using the GLO approximation, we can simplify reachability analysis as follows:

Theorem 1 (Reachability using the GLO approximation). The reachable set of outputs τ_k for test case m can be overapproximated as

$$\hat{\mathcal{Y}}_k^{(m)} = \bar{\mathcal{Y}}_k^{(m)} \oplus \hat{\mathcal{E}}_k^{(m)}, \quad (6)$$

where

$$\begin{aligned} \bar{\mathcal{Y}}_k^{(m)} &= \bar{y}_k^{(m)} \oplus \bar{C}_k^{(m)}(-c_x \oplus \mathcal{X}_0^{(m)}) \\ &\quad \oplus \bigoplus_{i=0}^k \bar{D}_{k,i}^{(m)}(-c_u \oplus \mathcal{U}_i^{(m)}), \end{aligned} \quad (7)$$

and $\bar{y}_k^{(m)}$, $\bar{C}_k^{(m)}$, and $\bar{D}_{k,i}^{(m)}$ are the reference output, system matrix, and input matrices of the GLO approximation of τ_k along the reference trajectory defined by

$$\bar{x}_0^{(m)} = x_{*,0}^{(m)} + c_x, \quad (8a)$$

$$\bar{u}_i^{(m)} = u_{*,i}^{(m)} + c_u, \quad i = 0, \dots, k. \quad (8b)$$

The vectors c_x and c_u are the initial estimates for the center vectors of the uncertainty sets $\mathcal{X}_0^{(m)}$ and $\mathcal{U}_i^{(m)}$, and the set $\hat{\mathcal{E}}_k^{(m)}$ is an enclosure of the linearization errors $\bar{e}_k^{(m)}$. The overall tightness of the overapproximation $\hat{\mathcal{Y}}_k^{(m)} \supseteq \mathcal{Y}_k^{(m)}$ depends solely on the conservatism introduced by $\hat{\mathcal{E}}_k^{(m)}$.

Proof. Using Def. 7, the output for test case m can be expressed as

$$\tau_k(x_0, u_{0:k}) = \bar{\tau}_k^{(m)}(x_0, u_{0:k}) + \bar{e}_k^{(m)}(x_0, u_{0:k}). \quad (9)$$

A set-based evaluation of (9) over the uncertain initial state $x_0 \in x_{*,0} \oplus \mathcal{X}_0^{(m)}$ and the uncertain inputs $u_i \in u_{*,i}^{(m)} \oplus \mathcal{U}_i^{(m)}$, $i = 0, \dots, k$, leads to (6), where the reachable set of the GLO approximation

$$\{\bar{\tau}_k^{(m)}(x_0, u_{0:k}) \mid x_0 \in x_{*,0}^{(m)} \oplus \mathcal{X}_0^{(m)}, u_{0:k} \in u_{*,0:k}^{(m)} \oplus \mathcal{U}_{0:k}^{(m)}\}$$

is denoted by $\bar{\mathcal{Y}}_k^{(m)}$, and the set $\hat{\mathcal{E}}_k^{(m)}$ encloses all linearization errors. Eq. (7) can be obtained from a set-based evaluation of (5) and the reference trajectory in (8) centers the linearization at the expected midpoint of the initial state and input sets to reduce the expected linearization error.

Although the center shifts $c_{\Delta,x}$ and $c_{\Delta,u}$ are unknown, they are treated as point values. This modeling choice implies that dependencies between uncertainties at different time steps are not considered (i.e., each uncertain variable takes values from its respective uncertainty set independently of the others), thereby avoiding any additional overapproximation that could arise from ignoring such dependencies [39]. Moreover, since the uncertain initial state and all uncertain input variables appear only once in the GLO approximation, we do not encounter the dependency effect [40, Sec. 2.2.3] in (7), which would otherwise introduce additional conservatism. As a result, (7) computes the exact reachable set of the GLO approximation, and the overall tightness of $\hat{\mathcal{Y}}_k^{(m)}$ depends solely on the conservatism introduced by $\hat{\mathcal{E}}_k^{(m)}$. $\square$

Using this theorem, we can compute the reachable set for different model types, where we compute the GLO approximation $\bar{\tau}_k^{(m)}$ from a first-order Taylor series expansion of the output τ_k along the reference trajectory to obtain small linearization errors. In the following subsections, we present the formulas for computing the GLO approximation and the set of linearization errors for state-space and input-output models. For improved readability, we omit the test case m in the superscripts in the following derivations.

A. State-Space Models

A popular model type, which is especially used in control applications, is the state-space model. In the general nonlinear case, it is described by the model functions $\mathcal{F} = \{f, g\}$, where

$$x_{k+1} = f(x_k, u_k) \quad (10a)$$

$$y_k = g(x_k, u_k). \quad (10b)$$

Corollary 1 (Reachability of State-Space Models). *The reachable set of the state-space model in (10) can be computed using*

Thm. 1, with

$$\bar{y}_k = g(\bar{x}_k, \bar{u}_k), \quad (11a)$$

$$\bar{C}_k = C_k \prod_{l=1}^k A_{k-l}, \quad (11b)$$

$$\bar{D}_{k,i} = \begin{cases} C_k \left(\prod_{l=1}^{k-i-1} A_{k-l} \right) B_i & 0 \leq i < k, \\ D_k & i = k, \end{cases} \quad (11c)$$

$$\hat{\mathcal{E}}_k = C_k \bigoplus_{j=0}^{k-1} \left(\prod_{l=1}^{k-j-1} A_{k-l} \right) \hat{\mathcal{L}}_{x,j} \oplus \hat{\mathcal{L}}_{y,k}.$$

The reference output in (11a) is obtained from simulating (10) along the reference trajectory in (8), the sets $\hat{\mathcal{L}}_{x,j}$ and $\hat{\mathcal{L}}_{y,k}$ can be computed as in [41, Sec. 3.1], and

$$A_k = \nabla_{x_k} f(x_k, u_k)|_{\bar{x}_k, \bar{u}_k}, \quad B_k = \nabla_{u_k} f(x_k, u_k)|_{\bar{x}_k, \bar{u}_k}, \\ C_k = \nabla_{x_k} g(x_k, u_k)|_{\bar{x}_k, \bar{u}_k}, \quad D_k = \nabla_{u_k} g(x_k, u_k)|_{\bar{x}_k, \bar{u}_k}.$$

Proof. Applying a Taylor expansion of order κ with Lagrange remainder to (10) along the reference trajectory, we obtain

$$x_{k+1} = f(\bar{x}_k, \bar{u}_k) + A_k(x_k - \bar{x}_k) + B_k(u_k - \bar{u}_k) + l_{x,k} \quad (12)$$

$$y_k = g(\bar{x}_k, \bar{u}_k) + C_k(x_k - \bar{x}_k) + D_k(u_k - \bar{u}_k) + l_{y,k}, \quad (13)$$

where $l_{x,k} \in \mathcal{L}_{x,k}$ and $l_{y,k} \in \mathcal{L}_{y,k}$ contain the Taylor terms from order two up to order κ and the Lagrange remainders [41]. Recursively inserting (12) in (13) leads to the GLO approximation defined by the formulas above with

$$\bar{e}_k = C_k \sum_{j=0}^{k-1} \left(\prod_{l=1}^{k-j-1} A_{k-l} \right) l_{x,j} + l_{y,k}.$$

The set $\hat{\mathcal{E}}_k$ enclosing the linearization errors is computed by a set-based evaluation of $\bar{e}_k$ using the overapproximations $\hat{\mathcal{L}}_{x,j} \supseteq \mathcal{L}_{x,j}$ and $\hat{\mathcal{L}}_{y,k} \supseteq \mathcal{L}_{y,k}$ from [41, Sec. 3.1]. $\square$

If the state-space model is linear, we have $\hat{\mathcal{E}}_k = \mathbf{0}$.

B. Input-Output Models

To compute the reachable set of state-space models, we require some estimate for the initial state $x_{*,0}$, which is often not directly measurable. In that case, guaranteed state estimation [42] is required, which introduces additional uncertainties. By contrast, input-output models can be directly initialized from measurements, avoiding the need for state estimation and often resulting in tighter reachable sets [43]. This makes input-output models a valuable alternative for data-driven system identification.

A widely used class of input-output models is the nonlinear autoregressive model with exogenous inputs (NARX model), where the current output is computed from the current input and a finite history of past outputs and inputs using the model function $\mathcal{F} = \{h\}$:

$$y_k = h(y_{k-n_p:k-1}, u_{k-n_p:k}), \quad (14)$$

and n_p denotes the number of past time steps considered. The model can be recursively evaluated from initial

measurements, which form the initial state vector $x_0 = [y_0^\top \ y_1^\top \ \cdots \ y_{n_p-1}^\top]^\top$. By combining a Taylor expansion with the reformulation from [43, Thm. 1], we obtain the following formulas to compute the reachable set:

Corollary 2 (Reachability of Input-Output Models). *The reachable set of the input-output model in (14) can be computed using Thm. 1, with $k \geq n_p$ and*

$$\bar{y}_k = h(\bar{y}_{k-n_p:k-1}, \bar{u}_{k-n_p:k}), \quad (15a)$$

$$\bar{C}_k = E \prod_{l=0}^{k-n_p} A_{\text{ext},k-l}, \quad (15b)$$

$$\bar{D}_{k,i} = E \sum_{j=0}^{k-n_p} \left(\prod_{l=0}^{j-1} A_{\text{ext},k-l} \right) B_{\text{ext},k-j,k-i-j}, \quad (15c)$$

$$\hat{\mathcal{E}}_k = E \bigoplus_{j=0}^{k-n_p} \left(\prod_{l=0}^{j-1} A_{\text{ext},k-l} \right) E^\top \hat{\mathcal{L}}_{k-j}.$$

The reference output in (15a) is obtained by simulating (14) along the reference trajectory in (8), the set $\hat{\mathcal{L}}_k$ can be computed analogously to [41, Sec. 3.1], and

$$\begin{aligned} E &= [\mathbf{0}_{n_y \times (n_p-1)n_y} \quad \mathbf{I}_{n_y}], \\ A_{\text{ext},k} &= \begin{bmatrix} \mathbf{0}_{(n_p-1)n_y \times n_y} & \mathbf{I}_{(n_p-1)n_y} \\ A_{k,n_p} & [A_{k,n_p-1} \ \cdots \ A_{k,1}] \end{bmatrix}, \\ B_{\text{ext},k,i} &= \begin{bmatrix} \mathbf{0}_{(n_p-1)n_y \times n_u} \\ B_{k,i} \end{bmatrix}, \\ A_{k,i} &= \nabla_{y_{k-i}} h(y_{k-n_p:k-1}, u_{k-n_p:k})|_{\bar{y}_{k-n_p:k-1}, \bar{u}_{k-n_p:k}}, \\ B_{k,i} &= \nabla_{u_{k-i}} h(y_{k-n_p:k-1}, u_{k-n_p:k})|_{\bar{y}_{k-n_p:k-1}, \bar{u}_{k-n_p:k}}. \end{aligned}$$

Proof. The output of the NARX model can be approximated by a Taylor series of order κ along the reference trajectory defined by $\bar{x}_0 = [\bar{y}_0^\top \ \bar{y}_1^\top \ \cdots \ \bar{y}_{n_p-1}^\top]^\top$ and $\bar{u}_{0:k}$:

$$\begin{aligned} y_k &= h(\bar{y}_{k-n_p:k-1}, \bar{u}_{k-n_p:k}) + \sum_{i=1}^{n_p} A_{k,i} (y_{k-i} - \bar{y}_{k-i}) \\ &\quad + \sum_{i=0}^{n_p} B_{k,i} (u_{k-i} - \bar{u}_{k-i}) + l_k, \end{aligned}$$

where $l_k \in \mathcal{L}_k$ contains the Taylor terms from order two up to order κ and the Lagrange remainder [41]. A reformulation analog to [43, Thm. 1] results in the GLO approximation defined by the formulas above with

$$\bar{e}_k = E \sum_{j=0}^{k-n_p} \left(\prod_{l=0}^{j-1} A_{\text{ext},k-l} \right) E^\top l_{k-j}.$$

A set-based evaluation of $\bar{e}_k$ leads to the set $\hat{\mathcal{E}}_k$ using the overapproximation $\hat{\mathcal{L}}_k \supseteq \mathcal{L}_k$ from [41, Sec. 3.1]. $\square$

If the function h in (14) is linear in $y_{k-n_p:k-1}$ and $u_{k-n_p:k}$, i.e., the model is an ARX model, we have $\hat{\mathcal{E}}_k = \mathbf{0}$.

IV. IDENTIFICATION OF WHITE-BOX MODELS

This section describes the general framework to identify uncertainty sets that ensure reachset conformance for models with known model functions $\mathcal{F}$ (Problem 1.1). For linear

models and the zonotopic uncertainty sets in (3), reachset conformance can be established by computing scaling factors α and center shifts c_Δ with linear programming [32]. This approach can be generalized to a nonlinear model by computing its reachable set using Thm. 1 for each test case m . To formulate the white-box identification problem from Problem 1.1 as a linear program (LP), we require the following lemmas.

Lemma 1 (Set of Output Deviations). *Given the uncertainty sets in (3), the center vector and generators of the set $\bar{\mathcal{Y}}_k^{(m)}$ from Thm. 1 are*

$$\text{cen}(\bar{\mathcal{Y}}_k^{(m)}) = \bar{y}_k^{(m)} + \bar{C}_k^{(m)} c_{\Delta,x} + \sum_{i=0}^k \bar{D}_{k,i}^{(m)} c_{\Delta,u}, \quad (16a)$$

$$\text{gen}(\bar{\mathcal{Y}}_k^{(m)}) = \text{gen}'(\bar{\mathcal{Y}}_k^{(m)}) \text{diag} \left([\alpha_x^\top \ \alpha_u^\top \ \cdots \ \alpha_u^\top]^\top \right), \quad (16b)$$

$$\text{gen}'(\bar{\mathcal{Y}}_k^{(m)}) = \begin{bmatrix} \bar{C}_k^{(m)} G_x & \bar{D}_{k,0}^{(m)} G_u & \cdots & \bar{D}_{k,k}^{(m)} G_u \end{bmatrix}. \quad (16c)$$

Proof. We insert the uncertainty sets from (3) in (7). $\square$

The cost function in (4a) can be expressed linearly in α as follows:

Lemma 2 (Cost Function). *Given Lemma 1 and using $r(\hat{\mathcal{Y}}_k^{(m)}) = \|\bar{\mathcal{Y}}_k^{(m)}\|_I$, the cost in (4a) can be computed as $\gamma\alpha$, where $\alpha \geq \mathbf{0}$ and*

$$\gamma = \sum_{m \in \mathcal{M}} \sum_{k=0}^{n_k-1} w_k \mathbf{1}^\top \left[|\bar{C}_k^{(m)} G_x| \sum_{i=0}^k |\bar{D}_{k,i}^{(m)} G_u| \right]. \quad (17)$$

Proof. We evaluate (4a), where $\|\bar{\mathcal{Y}}_k^{(m)}\|_I$ is computed by inserting (16b) into Def. 3, analogously to [32, Lem. 1]. $\square$

To enforce the conformance constraints in (4b), we use one of the following lemmas, which are based on the halfspace representation or the generator representation of $\bar{\mathcal{Y}}_k^{(m)}$.

Lemma 3 (Halfspace Constraints). *Given Lemma 1, the containment constraint $y_k^{(m,s)}(S_T) \in \bar{\mathcal{Y}}_k^{(m)}$, $\forall s$, is equivalent to*

$$\max_s \left(N_k^{(m)} y_{a,k}^{(m,s)} \right) \leq P_{\alpha,k}^{(m)} \alpha + P_{c,k}^{(m)} c_\Delta, \quad (18)$$

with $\alpha \geq \mathbf{0}$, $y_{a,k}^{(m,s)} = y_k^{(m,s)}(S_T) - \bar{y}_k^{(m)}$, and

$$P_{\alpha,k}^{(m)} = \left[|N_k^{(m)} \bar{C}_k^{(m)} G_x| \sum_{i=0}^k |N_k^{(m)} \bar{D}_{k,i}^{(m)} G_u| \right],$$

$$P_{c,k}^{(m)} = \left[N_k^{(m)} \bar{C}_k^{(m)} \quad N_k^{(m)} \sum_{i=0}^k \bar{D}_{k,i}^{(m)} \right].$$

The rows of $N_k^{(m)}$ are the normal vectors of the halfspace representation of $\bar{\mathcal{Y}}_k^{(m)}$, which can be obtained using [36, Thm. 7].

Proof. By subtracting $\bar{y}_k^{(m)}$ on both sides, we can write the constraint $y_k^{(m,s)}(S_T) \in \bar{\mathcal{Y}}_k^{(m)}$ as $y_{a,k}^{(m,s)} \in -\bar{y}_k^{(m)} \oplus \bar{\mathcal{Y}}_k^{(m)}$. With the halfspace representation of $\bar{\mathcal{Y}}_k^{(m)}$, where $\bar{\mathcal{Y}}_k^{(m)}$ can

be computed using Lemma 1, this constraint can be enforced for each test case m analogously to [32, Thm. 1]. $\square$

Lemma 4 (Generator Constraints). *Given Lemma 1, the containment constraint $y_k^{(m,s)}(S_T) \in \bar{\mathcal{Y}}_k^{(m)}$, $\forall s$, is equivalent to*

$$\forall s: \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \leq \begin{bmatrix} R_{\alpha,k}\alpha + \beta_k^{(m,s)} \\ R_{\alpha,k}\alpha - \beta_k^{(m,s)} \end{bmatrix}, \quad (19a)$$

$$y_{a,k}^{(m,s)} = Q_{c,k}^{(m)} c_{\Delta} + Q_{\beta,k}^{(m)} \beta_k^{(m,s)}, \quad (19b)$$

with $\alpha \geq \mathbf{0}$, $y_{a,k}^{(m,s)} = y_k^{(m,s)}(S_T) - \bar{y}_k^{(m)}$, and the auxiliary optimization variables $\beta_k^{(m,s)} \in \mathbb{R}^{\eta_x + (k+1)\eta_u}$. The constraint matrices are

$$Q_{c,k}^{(m)} = \begin{bmatrix} \bar{C}_k^{(m)} & \sum_{i=0}^k \bar{D}_{k,i}^{(m)} \end{bmatrix},$$

$$Q_{\beta,k}^{(m)} = \text{gen}'(\bar{\mathcal{Y}}_k^{(m)}),$$

$$R_{\alpha,k} = \begin{bmatrix} \mathbf{I}_{\eta_x} & \mathbf{0}_{\eta_x \times \eta_u} \\ \mathbf{0}_{(k+1)\eta_u \times \eta_x} & \text{vert}_{k+1}(\mathbf{I}_{\eta_u}) \end{bmatrix},$$

where $\text{vert}_{k+1}(\mathbf{I}_{\eta_u})$ is the vertical concatenation of $k+1$ identity matrices and $\text{gen}'(\bar{\mathcal{Y}}_k^{(m)})$ is computed as in (16c).

Proof. We can write the constraint $y_k^{(m,s)}(S_T) \in \bar{\mathcal{Y}}_k^{(m)}$ as $y_{a,k}^{(m,s)} \in -\bar{y}_k^{(m)} \oplus \bar{\mathcal{Y}}_k^{(m)}$ after subtracting $\bar{y}_k^{(m)}$ on both sides. This constraint can be enforced via the generator representation of $\bar{\mathcal{Y}}_k^{(m)}$ analogously to [32, Thm. 2], i.e.,

$$y_{a,k}^{(m,s)} \in -\bar{y}_k^{(m)} \oplus \bar{\mathcal{Y}}_k^{(m)}$$

$$\stackrel{\text{Def. 1}}{\Leftrightarrow} \exists \lambda_k^{(m,s)} \in \mathbb{R}^{\eta_x + (k+1)\eta_u} :$$

$$|\lambda_k^{(m,s)}| \leq \mathbf{1},$$

$$y_{a,k}^{(m,s)} = -\bar{y}_k^{(m)} + \text{cen}(\bar{\mathcal{Y}}_k^{(m)}) + \text{gen}(\bar{\mathcal{Y}}_k^{(m)})\lambda_k^{(m,s)}$$

$$\stackrel{(16)}{\Leftrightarrow} \exists \beta_k^{(m,s)} \in \mathbb{R}^{\eta_x + (k+1)\eta_u} :$$

$$|\beta_k^{(m,s)}| \leq [\alpha_x^\top \quad \alpha_u^\top \quad \cdots \quad \alpha_u^\top]^\top, \quad (21a)$$

$$y_{a,k}^{(m,s)} = \bar{C}_k^{(m)} c_{\Delta,x} + \sum_{i=0}^k \bar{D}_{k,i}^{(m)} c_{\Delta,u}$$

$$+ \text{gen}'(\bar{\mathcal{Y}}_k^{(m)})\beta_k^{(m,s)}, \quad (21b)$$

where we define $\beta_k^{(m,s)} := \text{diag}([\alpha_x^\top \quad \alpha_u^\top \quad \cdots \quad \alpha_u^\top]^\top)\lambda_k^{(m,s)}$. Finally, the constraints in (21a) and (21b) can be written as in (19a) and (19b). $\square$

The resulting white-box identification procedure is summarized in Alg. 1. It takes as input the set of model functions $\mathcal{F}$ (i.e., $\mathcal{F} = \{f, g\}$ from (10) for state-space models or $\mathcal{F} = \{h\}$ from (14) for input-output models), the matrix B_c that determines which components of the center shift c_{Δ} are to be identified, and the set of test cases $\mathcal{M}$. In Line 1, we compute the GLO approximations $\bar{\tau}^{(m)}$ for all $m \in \mathcal{M}$ using the function `computeGLO()`, which implements (11) for state-space models or (15) for input-output models. The scaling factors α and center shifts c_{Δ} are then identified by solving the linear program in (22), as shown in Line 6. This optimization minimizes the cost function defined in Lemma 2, subject to the reachset-conformance constraints provided by

Algorithm 1 White.

Inputs:

$\mathcal{F}$ Model functions
 B_c Center identification matrix
 $\mathcal{M}$ Test cases

Outputs:

α^* Optimized scaling factors
 c_{Δ}^* Optimized center shifts
 ℓ_C^* Final conformance cost

Compute GLO approximation for each test case

1: $\bar{\tau}^{(m)} \leftarrow \text{computeGLO}(\mathcal{F}, m), \forall m \in \mathcal{M}$

Optimize scaling factors and center shifts

2: $\gamma \leftarrow (17)$

Ensure B_c is identity matrix for nonlinear model functions

3: **if** any function in $\mathcal{F}$ is nonlinear **then**

4: $B_c \leftarrow \mathbf{I}$

5: **end if**

6: $\alpha^*, c_{\Delta}^* \leftarrow \arg \min_{\alpha, c_{\Delta}, (\beta)} \gamma \alpha$ (22a)

s.t. constraint (18) or (19), $\forall m, k$, (22b)

$\alpha \geq \mathbf{0}$, (22c)

$B_c c_{\Delta} = \mathbf{0}$, (22d)

Compute the final conformance cost

7: $\ell_C^* \leftarrow \gamma \alpha^*$

Lemma 3 or Lemma 4. The matrix B_c , a diagonal matrix with entries in $\{0, 1\}$, encodes which elements of c_{Δ} are fixed to zero via the constraint in (22d). Since estimating optimal center shifts for nonlinear models would require introducing nonlinear constraints (see the proof of Thm. 2), we enforce $c_{\Delta} = \mathbf{0}$ for such models by setting B_c to the identity matrix in Lines 3 to 5. To identify the center vectors for nonlinear models, we can consider c as a part of the unknown model parameters p and use the methods presented in the next section.

Theorem 2 (White-Box Identification of Uncertainty Sets). *Problem 1.1 can be solved with Alg. 1.*

Proof. If the sets of linearization errors $\hat{\mathcal{E}}_k^{(m)}$ used in (6) contain the zero vectors, i.e., $\mathbf{0} \in \hat{\mathcal{E}}_k^{(m)}$, $k = 0, \dots, n_k - 1$, then $\hat{\mathcal{Y}}_k^{(m)}$ is underapproximated by $\bar{\mathcal{Y}}_k^{(m)}$, which can be computed using (7). By using this underapproximation and defining the function $r(\hat{\mathcal{Y}}_k^{(m)}) = \|\bar{\mathcal{Y}}_k^{(m)}\|_I$ as in Lemma 2, Problem 1.1 can be simplified to

$$\arg \min_{\alpha, c_{\Delta}} \gamma \alpha \quad (23a)$$

$$\text{s.t. } \forall m, k, s: y_k^{(m,s)}(S_T) \in \bar{\mathcal{Y}}_k^{(m)}. \quad (23b)$$

Feasible solutions to this optimization problem lead to reachset-conformant models since the satisfaction of the more restrictive constraint (23b) implies the satisfaction of (4b). According to Lemma 3 and Lemma 4, (23b) can be expressed with the linear constraints given in (22b) and (22c).

The requirement $\mathbf{0} \in \hat{\mathcal{E}}_k^{(m)}$ is trivially fulfilled for linear systems. For nonlinear systems, this condition is satisfied by Corollary 1 and Corollary 2 if the initial state and inputs of

Algorithm 2 GraySim.**Inputs:**

p Initial estimate for the model parameters
 $\mathcal{F}_p$ Model functions parameterized by p
 B_c Center identification matrix
 $\mathcal{M}$ Test cases

Outputs:

p^* Optimized model parameters
 α^* Optimized scaling factors
 c_Δ^* Optimized center shifts
 ℓ_C^* Final conformance cost

Find parameter p that minimize the cost of the white-box identification

```
1: while NLP solver has not terminated do
2:    $\alpha, c_\Delta, \ell_C \leftarrow \text{White}(\mathcal{F}_p, B_c, \mathcal{M})$ 
3:    $p \leftarrow \text{next NLP-iteration to minimize } \ell_C$ 
4: end while
5:  $p^* \leftarrow p$ 
```

*Optimize scaling factors, center shifts, and cost given p^**

```
6:  $\alpha^*, c_\Delta^*, \ell_C^* \leftarrow \text{White}(\mathcal{F}_{p^*}, B_c, \mathcal{M})$ 
```

the reference trajectory $(\bar{x}_0^{(m)}, \bar{u}_{0:k}^{(m)})$, which are used for the linearization, are contained in the initial state and input sets, respectively, i.e., $\bar{x}_0^{(m)} \in x_{*,0}^{(m)} \oplus \mathcal{X}_0^{(m)}$ and $\bar{u}_i^{(m)} \in u_{*,i}^{(m)} \oplus \mathcal{U}_i$, $\forall i$. Since $x_{*,0}^{(m)} \oplus \mathcal{X}_0^{(m)}$ and $u_{*,i}^{(m)} \oplus \mathcal{U}_i^{(m)}$ have their centers at $\bar{x}_0^{(m)} + c_{\Delta,x}$ and $\bar{u}_i^{(m)} + c_{\Delta,u}$, this containment can be satisfied by setting $c_\Delta = [c_{\Delta,x}^\top, c_{\Delta,u}^\top]^\top$ to the zero vector with (22d). $\square$

V. IDENTIFICATION OF GRAY-BOX MODELS

In this section, we extend the uncertainty identification approach from Thm. 2 to the reachset-conformant identification of gray-box models (Problem 1.2). First, we describe an algorithm that integrates the LP from Sec. IV in a nonlinear program (NLP) to simultaneously identify the uncertainty sets and the unknown model parameters. As an alternative with lower computational complexity, we present a sequential identification approach, which identifies the model parameters prior to the uncertainty estimation.

The uncertainty sets and unknown model parameters p can be simultaneously identified with the approach proposed in [44, Sec. III-B], termed GraySim. The general procedure is provided in Alg. 2. The function `White` implements Alg. 1 and returns scaling factors α , center shifts c_Δ , and the conformance cost ℓ_C for the model functions $\mathcal{F}_p$. The outer nonlinear program uses ℓ_C to optimize the parameters p in Lines 1 to 4 until the solver converges or exceeds a maximum number of iterations. While we do not have any guarantees for finding the globally optimal parameters, we can ensure that the final model is reachset-conformant by computing the scaling factors α^* and center shifts c_Δ^* with `White` for the final parameters p^* in Line 6.

Solving the LP in `White` at each iteration of the NLP, as proposed in [44], leads to high computational complexity. Furthermore, only NLP solvers that do not require a closed-form expression of the cost function can be used. Thus,

Algorithm 3 GraySeq.**Inputs:**

p Initial estimate for the model parameters
 $\mathcal{F}_p$ Model functions parameterized by p
 B_c Center identification matrix
 $\mathcal{M}$ Test cases

Outputs:

p^* Optimized model parameters
 α^* Optimized scaling factors
 c_Δ^* Optimized center shifts
 ℓ_C^* Final conformance cost

Find parameter p that minimize the cost function from Prop. 1

```
1: while NLP solver has not terminated do
   Compute GLO approximation for each test case
2:    $\bar{\tau}^{(m)} \leftarrow \text{computeGLO}(\mathcal{F}_p, m), \forall m \in \mathcal{M}$ 
3:    $\ell_U \leftarrow (24)$ 
4:    $p, c_\Delta \leftarrow \text{next NLP-iteration to minimize } \ell_U$ 
5: end while
6:  $p^* \leftarrow p$ 
   Optimize scaling factors, center shifts, and cost given  $p^*$ 
7:  $\alpha^*, c_\Delta^*, \ell_C^* \leftarrow \text{White}(\mathcal{F}_{p^*}, B_c, \mathcal{M})$ 
```

we propose Alg. 3, termed GraySeq, resulting in shorter computation times and higher flexibility in the choice of the NLP solver. First, we identify model parameters p^* and center shifts c_Δ in Lines 1 to 6 by minimizing an approximation of the conformance cost ℓ_C , which is independent of the scaling factors α . Next, we run the white-box identification algorithm `White` using the optimized parameters p^* to compute the uncertainty sets and the conformance cost in Line 7.

We propose to use the following cost function for the estimation of the parameters p :

Proposition 1 (Underapproximative Cost Function). *The minimum cost $\gamma\alpha^*$ of the LP in (22) can be underapproximated by*

$$\ell_U = \sum_{m \in \mathcal{M}} \sum_{k=0}^{n_k-1} w_k \mathbf{1}^\top \max_s |y_{a,k}^{(m,s)} - y_{\Delta,k}^{(m)}|, \quad (24)$$

with

$$\begin{aligned} y_{a,k}^{(m,s)} &= y_k^{(m,s)}(S_T) - \bar{y}_k^{(m)}, \\ y_{\Delta,k}^{(m)} &= \left[\bar{C}_k^{(m)} \quad \sum_{i=0}^k \bar{D}_{k,i}^{(m)} \right] \tilde{B}_c c_\Delta, \\ \tilde{B}_c &= \begin{cases} \mathbf{0} & \text{if any function in } \mathcal{F}_p \text{ is nonlinear,} \\ |B_c - \mathbf{I}| & \text{else.} \end{cases} \end{aligned}$$

Proof. Given Lemma 4, we have $|\beta_k^{(m,s)}| \leq \alpha_k^*$ (see (21a)) with $\alpha_k^* := [\alpha_x^{*\top} \alpha_u^{*\top} \cdots \alpha_u^{*\top}]^\top$ and

$$y_{a,k}^{(m,s)} - \bar{C}_k^{(m)} c_{\Delta,x} - \sum_{i=0}^k \bar{D}_{k,i}^{(m)} c_{\Delta,u} = \text{gen}'(\bar{y}_k^{(m)}) \beta_k^{(m,s)}.$$

By taking the absolute value of both sides, we obtain

$$\begin{aligned} |y_{a,k}^{(m,s)} - \bar{C}_k^{(m)} c_{\Delta,x} - \sum_{i=0}^k \bar{D}_{k,i}^{(m)} c_{\Delta,u}| &= |\text{gen}'(\bar{\mathcal{Y}}_k^{(m)}) \beta_k^{(m,s)}| \\ &\leq |\text{gen}'(\bar{\mathcal{Y}}_k^{(m)})| |\beta_k^{(m,s)}| \\ &\leq |\text{gen}'(\bar{\mathcal{Y}}_k^{(m)})| \alpha_k^*. \end{aligned}$$

As this has to hold for all s , we can take the maximum over s , resulting in

$$\max_s |y_{a,k}^{(m,s)} - \bar{C}_k^{(m)} c_{\Delta,x} - \sum_{i=0}^k \bar{D}_{k,i}^{(m)} c_{\Delta,u}| \leq |\text{gen}'(\bar{\mathcal{Y}}_k^{(m)})| \alpha_k^*.$$

At the same time, the minimum cost of (22) can be computed as

$$\ell_C^* = \sum_{m \in \mathcal{M}} \sum_{k=0}^{n_k-1} w_k \mathbf{1}^\top |\text{gen}'(\bar{\mathcal{Y}}_k^{(m)})| \alpha_k^*.$$

Combining the last two formulas and considering constraint (22d) leads to the underapproximation $\ell_U \leq \gamma \alpha^*$. $\square$

Although we do not have any guarantees on the closeness of the conformance cost ℓ_C and the underapproximation ℓ_U , this cost function works well in practice as demonstrated in Sec. VII. Alternatively, we can use the least-squares cost function

$$\ell_{LS} = \sum_{m \in \mathcal{M}} \sum_{k=0}^{n_k-1} w_k \mathbf{1}^\top \sum_{s=1}^{n_s} \left(y_{a,k}^{(m,s)} - y_{\Delta,k}^{(m)} \right)^2. \quad (25)$$

This cost function decreases the computation time even more, but the results are more vulnerable to not centrally distributed uncertainties compared to using cost functions that consider only the maximum errors. A detailed analysis of the accuracy and computation times is presented in Sec. VII-B.

VI. IDENTIFICATION OF BLACK-BOX MODELS

For the black-box identification of reachset-conformant models (Problem 1.3), we propose a sequential approach that first identifies the model functions followed by the computation of the uncertainty sets using the methods from Sec. IV. We use genetic programming (GP) in this work, but other black-box identification methods, such as the training of neural networks, can be used similarly [45]–[48].

GP is a global optimization technique that iteratively evolves a population of symbolic functions or entire computer programs [49]. The evolution is driven by the need to optimize a fitness function mimicking the Darwinian principle of survival of the fittest. GP for system identification can be described by the following steps:

- 1) **Initialization:** A population of random symbolic functions, called individuals, is generated as a weighted sum of different genes. Each gene is a nonlinear function represented by a tree structure, constructed using a specified set of building block functions, input variables, and optionally, random constants. An example gene is displayed in Fig. 2.

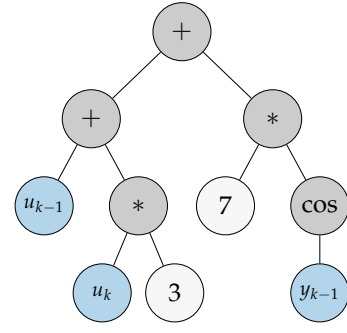


Fig. 2. Example gene in GP using the building blocks $+$, $*$, and $\cos$, the input variables u_k , u_{k-1} , and y_{k-1} , and the random constants 3 and 7 .

- 2) **Fitness Evaluation:** The fitness function evaluates the fit of each individual to the training data.
- 3) **Evolution:** Individuals are selected based on their fitness alone or on both their fitness and their complexity. The selected individuals undergo the genetic operations crossover (genes are exchanged between individuals), mutation (random changes to individuals to maintain genetic diversity), and replication (some high-performing individuals are copied directly to the next generation). The resulting individuals are part of the next generation, where we continue with step 2.
- 4) **Termination:** The evolution process is terminated if a fitness threshold is exceeded or the maximum number of generations is reached.

Overfitting to the given data is mitigated by limiting the maximum number of nodes and the maximum depth of the trees, as well as penalizing complexity in the evolution process.

As some of the models identified with GP might lead to exploding reachable sets, even though we observed small prediction errors in the training process, we propose a new GP variant termed *conformant genetic programming (CGP)*. Exploding reachable sets can come from inherent model dynamics that were not observable for the test cases in the training dataset. For example, we might have a division by a near-zero number in the output equation for a particular input trajectory, which would lead to large outputs. The probability that this input trajectory is contained in the training data is small, but the probability is significantly higher when considering whole input sets. Thus, it is advantageous to consider the cost ℓ_C of (22), which describes the size of the reachable set for a conformant model, in the training process. However, since the computation of the conformance cost requires the solution of an LP, this cost leads to a significantly higher computational complexity than a least-squares cost function like (25) and can only be evaluated for small test suites in reasonable computation time. Hence, we propose the approach described in Alg. 4, where we first minimize a least-squares cost function, followed by a few iterations of minimizing the conformance cost. We use the following functions in the algorithm:

- **Initialize**, which randomly creates n_τ sets of model functions with different nonlinearities [50, Sec. 3.2],
- **Evolve**, which implements the evolution process based

Algorithm 4 BLackCGP.

Inputs:
 $\mathcal{M}_{T1}$ Training test cases for step 1
 $\mathcal{M}_{T2}$ Training test cases for step 2
 $\mathcal{M}_C$ Conformance test cases

Outputs:
 $\mathcal{F}^*$ Identified model functions
 α^* Optimized scaling factors
 c_Δ^* Optimized center shifts
 ℓ_C^* Final conformance cost

Step 1: Minimize least-squares cost
Initialize the function sets
1: $\{\mathcal{F}_1, \dots, \mathcal{F}_{n_f}\} \leftarrow \text{Initialize}$
2: **for** $i = \{1, \dots, n_g\}$ **do**
 Compute least-squares cost for each function set
3: **for** $j = \{1, \dots, n_f\}$ **do**
4: $\tau^{(m)} \leftarrow \text{computeGLO}(\mathcal{F}_j, m), \forall m \in \mathcal{M}_{T1}$
5: $\ell_j \leftarrow (25)$
6: **end for**
7: **if** $i < n_g$ **then**
 Optimization via genetic evolution
8: $\{\mathcal{F}_1, \dots, \mathcal{F}_{n_f}\} \leftarrow \text{Evolve}((\mathcal{F}_1, \ell_1), \dots, (\mathcal{F}_{n_f}, \ell_{n_f}))$
9: **end if**
10: **end for**
 Selection of the $\tilde{n}_f$ function sets with the lowest cost
11: $\{\mathcal{F}_1, \dots, \mathcal{F}_{\tilde{n}_f}\} \leftarrow \text{Best}((\mathcal{F}_1, \ell_1), \dots, (\mathcal{F}_{n_f}, \ell_{n_f}), \tilde{n}_f)$
Step 2: Minimize conformance cost
12: **for** $i = \{1, \dots, \tilde{n}_g\}$ **do**
 Compute sum of conformance costs for each function set
13: **for** $j = \{1, \dots, \tilde{n}_f\}$ **do**
14: $\ell_j \leftarrow 0$
15: **for** $l = \{1, \dots, n_v\}$ **do**
16: $\alpha, c_\Delta, \ell_C \leftarrow \text{White}(\mathcal{F}_j, \mathbf{0}, \mathcal{M}_{T2, l})$
17: $\ell_j \leftarrow \ell_j + \ell_C$
18: **end for**
19: **end for**
20: **if** $i < \tilde{n}_g$ **then**
 Optimization via genetic evolution
21: $\{\mathcal{F}_1, \dots, \mathcal{F}_{\tilde{n}_f}\} \leftarrow \text{Evolve}((\mathcal{F}_1, \ell_1), \dots, (\mathcal{F}_{\tilde{n}_f}, \ell_{\tilde{n}_f}))$
22: **end if**
23: **end for**
 Selection of the function set with the lowest cost
24: $\mathcal{F}^* \leftarrow \text{Best}((\mathcal{F}_1, \ell_1), \dots, (\mathcal{F}_{\tilde{n}_f}, \ell_{\tilde{n}_f}), 1)$
 Optimize scaling factors, center shifts, and cost given $\mathcal{F}^$*
25: $\alpha^*, c_\Delta^*, \ell_C^* \leftarrow \text{White}(\mathcal{F}^*, \mathbf{0}, \mathcal{M}_C)$

on the costs of the function sets [50, Sec. 3.2],

- **Best**, which returns the sets of model functions with the lowest cost, where the last argument specifies the number of returned sets, and
- **White**, which calls Alg. 1.

First, we randomly initialize a population of n_f sets of model functions, where the structure of each set $\mathcal{F}_j$ depends on the selected model class (e.g., $\mathcal{F}_j = \{h_j\}$ from (14) for input-output models). For the first n_g generations, we minimize the least-squares cost function for the training dataset $\mathcal{M}_{T1}$

to efficiently find good candidate functions (see Line 2 to Line 10). From the optimized function sets, we choose the $\tilde{n}_f$ sets with the lowest costs in Line 11. In the second part of the algorithm, we optimize the obtained functions by minimizing the conformance cost for the second training dataset $\mathcal{M}_{T2}$. To keep the computational complexity small, we split the dataset $\mathcal{M}_{T2}$ into the subdatasets $\mathcal{M}_{T2, l}$, $l = 1, \dots, n_v$, and compute the sum of conformance costs over all subdatasets for each function set in Lines 13 to 19. To activate the identification of center shifts in the rare case of linear model functions $\mathcal{F}_j$, we initialize B_c with the zero matrix (the algorithm **White** will automatically set B_c to the identity matrix for nonlinear models). Having reached the maximum number of generations $\tilde{n}_g$, we choose the function set with the lowest cost in Line 24 and compute the scaling factors and center shifts of a reachset-conformant model for the test cases in $\mathcal{M}_C$ in Line 25.

VII. NUMERICAL EXPERIMENTS

First, we evaluate the scalability of our white-box identification approach. Next, we analyze the general performance of the proposed white-, gray-, and black-box identification methods for different model types and compare them to baseline methods. Finally, we identify a reachset-conformant vehicle model from real-world measurements.

All methods are implemented in the MATLAB toolbox CORA [51] and are evaluated on an AMD EPYC 7763 processor with 2TB RAM and NVIDIA A100 40GB GPU. We use the built-in solvers **linprog** and **fmincon** for solving the required LPs and NLPs. Continuous-time models will be discretized for the identification using the forward Euler method.

A. Scalability of the White-Box Identification Approach

For evaluating the scalability of Alg. 1, we simulate a discretized version of a water tank system with uncertain inputs [52], whose dynamics are

$$\begin{aligned} \dot{x}_{t,1} &= u_{t,1} - 0.3\sqrt{x_{t,1}} \\ \dot{x}_{t,i} &= u_{t,i} + 0.3(\sqrt{x_{t,i-1}} - \sqrt{x_{t,i}}), \quad i = 2, \dots, n_x. \end{aligned}$$

The state $x_{t,i}$ describes the water level of tank i at time t , such that the system dimension n_x is equal to the number of tanks, and $u_{t,i}$ the water flow into tank i . We assume that there is an inflow in only every third tank, $u_{t,i} = 0$ for $i \neq 1, 4, 7, \dots$, and we measure each state variable, i.e., $y_{t,i} = x_{t,i}$ for $i = 1, \dots, n_y$. The computation times of the LP from Thm. 2 using the halfspace constraints from Lemma 3 and the generators constraints from Lemma 4 are presented for a) varying time horizons n_k , b) a varying number of test cases, which is denoted by n_m , c) a varying number of sample executions n_s of each test case, and d) a varying measurement dimension n_y . As we can see from Table I, increasing the number of samples n_s does not influence the computational complexity of the LP using the halfspace constraints since we only consider the sample $y_a^{(m,s)}$ closest to each halfspace in (18). However, as the halfspace conversion scales exponentially with the dimension of the reachable set [32], halfspace constraints can only be used for systems with

TABLE I
COMPUTATION TIMES FOR WHITE-BOX IDENTIFICATION.

n_k	n_m	n_s	n_y	Computation time of Thm. 2 [s]	
				Using Lemma 3	Using Lemma 4
4	10	1	3	0.16	0.15
20	10	1	3	16.14	0.83
40	10	1	3	>600.00	1.61
4	100	1	3	0.54	0.26
4	1000	1	3	23.26	2.58
4	10	10	3	0.16	0.18
4	10	100	3	0.16	1.43
4	10	1	6	9.84	0.22
4	10	1	9	>600.00	0.31
4	10	1	100	>600.00	8.53

small measurement dimension n_y and small time horizons n_k . In contrast, using the generator constraints is efficient for high-dimensional systems and long time horizons, but its computational complexity depends substantially on the overall number of measurement trajectories $n_s \cdot n_m$.

B. General Comparison for Different Model Types

We evaluate the proposed identification methods by computing reachset-conformant models for four dynamical systems, which are

- a pedestrian model [43] formulated as the linear state-space model

$$x_{k+1} = \begin{bmatrix} p_1 & 0 & p_2 & 0 \\ 0 & p_1 & 0 & p_2 \\ 0 & 0 & p_1 & 0 \\ 0 & 0 & 0 & p_1 \end{bmatrix} x_k + \begin{bmatrix} p_3 & 0 & 0 & 0 \\ 0 & p_3 & 0 & 0 \\ p_4 & 0 & 0 & 0 \\ 0 & p_4 & 0 & 0 \end{bmatrix} u_k,$$

$$y_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_k + \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} u_k,$$

where $p_1 = 1$, $p_2 = 0.01$, $p_3 = 5 \cdot 10^{-5}$, $p_4 = 0.01$,

- the pedestrian model converted to an ARX model, using the conversion formulas from [43, Prop. 2], where

$$y_k = \begin{bmatrix} p_1 & 0 \\ 0 & p_1 \end{bmatrix} y_{k-1} + \begin{bmatrix} p_2 & 0 \\ 0 & p_2 \end{bmatrix} y_{k-2}$$

$$+ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} u_k + \begin{bmatrix} p_3 & 0 & p_4 & 0 \\ 0 & p_3 & 0 & p_4 \end{bmatrix} u_{k-1}$$

$$+ \begin{bmatrix} p_3 & 0 & 1 & 0 \\ 0 & p_3 & 0 & 1 \end{bmatrix} u_{k-2},$$

with $p_1 = 2$, $p_2 = -2$, $p_3 = 5 \cdot 10^{-5}$, $p_4 = -1$,

- the Lorenz system [53] described by the nonlinear state-space model

$$\dot{x}_t = \begin{bmatrix} (p_1 + u_{t,1})(x_{t,2} - x_{t,1}) \\ (p_2 + u_{t,2})x_{t,1} - x_{t,2} - x_{t,1}x_{t,3} \\ x_{t,1}x_{t,2} - (p_3 + u_{t,3})x_{t,3} \end{bmatrix}$$

$$y_t = \begin{bmatrix} x_{t,1} \\ x_{t,2} \end{bmatrix},$$

with $p_1 = 10$, $p_2 = 28$, $p_3 = \frac{8}{3}$,

- a NARX model, which we call NARX1 and whose dynamics are adapted from [54]

$$y_k = \begin{bmatrix} \frac{y_{k-1,1}}{1 + y_{k-1,2}y_{k-1,2}} + p_1 u_{k-1,1} \\ \frac{y_{k-1,1}y_{k-1,2}}{1 + y_{k-1,2}y_{k-1,2}} + p_2 u_{k-2,2} \end{bmatrix},$$

with $p_1 = 0.8$, $p_2 = 1.2$.

The corresponding uncertainty sets $\mathcal{X}_0^{(m)}$ (only for the state-space models) and $\mathcal{U}_i^{(m)}$ are represented by zonotopes with a random center vector with elements between -1 and 1 and a random diagonal generator matrix with elements between -0.25 and 0.25 .

Using the true model, the true uncertainty sets, and random input trajectories, we simulate 100 conformance test suites $\mathcal{M}_C$ comprising $n_m = 20$ test cases each. Each test case has a length of $n_k = n_p + 6$ time steps and contains $n_s = 10$ measurement trajectories resulting from different uncertainty trajectories, where n_p is the model order for the ARX and NARX model (we require n_p measurements for their initialization) and zero for the state-space models if not defined otherwise. More data is required for the black-box identification approach as explained in the respective subsection. The matrices G_x and G_u for the uncertainty sets are initialized by identity matrices, and the initial estimates for the unknown parameters and center vectors are sampled from a Gaussian distribution with a standard deviation of 0.01.

We compare the identification methods using the following criteria:

- the *normalized cost*, which is the ratio of the final cost ℓ_C^* of the model identified with the considered approach and the cost using the true model and uncertainty sets,
- the *failure rate*, which is the ratio of the number of failure test suites, i.e., test suites where a conformant model could not be identified or where the normalized cost is bigger than 100, to the total number of test suites,
- the *computation time*, which is the time required for solving the conformance problem for one test suite, and
- the root mean squared errors (*RMSE*) between the true parameters and the parameters estimated with the gray-box identification approaches.

In Table II, we present the results averaged over all dynamical systems and non-failure test suites (except for the failure rate) for the proposed identification methods and baseline methods. Given the absence of comparable solutions in the existing literature, we benchmark our methods *White* from Alg. 1, *GraySeq* from Alg. 3, and *BlackCGP* from Alg. 4 against the following baseline approaches:

- *WhiteAdd*: This white-box identification approach is based on Alg. 1 but considers only additive measurement error sets in the optimization since previous research focused only on additive uncertainty sets [27], [30].
- *GraySim*: This gray-box identification approach implements the simultaneous identification scheme from Alg. 2, which was adapted from [44].
- *GraySeq2*: This gray-box identification approach is based on the sequential identification scheme of *GraySeq* in

TABLE II
EVALUATION OF THE PROPOSED IDENTIFICATION METHODS (IN **BOLD**)
AND BASELINE METHODS.

Approach	Normalized cost	Failure rate [%]	Computation time [s]	RMSE
White	1.00	0	0.09	-
WhiteAdd	1.88	0	0.10	-
GraySeq	1.02	4.00	12.56	0.19
GraySeq2	1.16	0.25	4.95	0.39
GraySim	1.49	11.50	20.67	0.24
BlackCGP	6.53	0.12	219.65	-
BlackGP	12.68	29.57	180.80	-

Alg. 3 but minimizes the least-squares cost function ℓ_{LS} from (25) with the NLP solver instead of ℓ_U .

- **BlackGP**: This black-box identification approach is based on the standard GP algorithm and aims to find a model minimizing the least-squares cost function in (25) before applying the white-box identification approach to find uncertainty sets that make the model reachset-conformant.

As the performance of the identification methods can vary substantially between different test suites and systems, we also visualize the normalized cost as boxplots for all non-failure test suites in Fig. 3. Example test cases, which were not contained in the identification data, consisting of $n_s = 50$ output trajectories with a length of $n_k = n_p + 10$ time steps, and the reachable sets from the identified models are displayed in Fig. 4. Since the GP approaches approximate the Lorenz model with a NARX model, which requires $n_p = 3$ measurements for its initialization, we enforce that all test cases visualized in Fig. 4c start with the same measurements $y_i, i = 0, \dots, n_p - 1$, such that we are able to compute one reachable set, which has to enclose all measurements. Thus, we start the reachability predictions at the time step $k = n_p$ and, to have a fair comparison, we use $x_{n_p} \oplus \mathcal{X}_0^{(m)}$ as the initial state set, which contains all states of the visualized test cases at $k = n_p$, for the nonlinear state-space models identified with the white and gray-box approaches.

1) **White-Box Identification**: We use Alg. 1 to identify the white-box models. Please keep in mind that the ARX and NARX models do not require to find α_x and that c_Δ is only identified for linear models. We use the halfspace constraints from Lemma 3 instead of the generator constraints from Lemma 4 to shorten computation times, because the considered systems are low-dimensional and the time horizon is short.

We can see from Table II that our identification algorithm, termed *White*, is able to estimate small uncertainty sets, which lead to tight reachable sets, in short computation times. Since the assumed directions for the generators were correct, linear programming is able to find a solution that is as good as using the true uncertainty sets, i.e., the normalized cost in Fig. 3 is always approximately equal to 1. Furthermore, Fig. 4 shows that the reachable set of the identified model is identical to the reachable set of the true system for the pedestrian state-space model and the pedestrian ARX model. Due to the linearization error, which is not considered in the identification but added

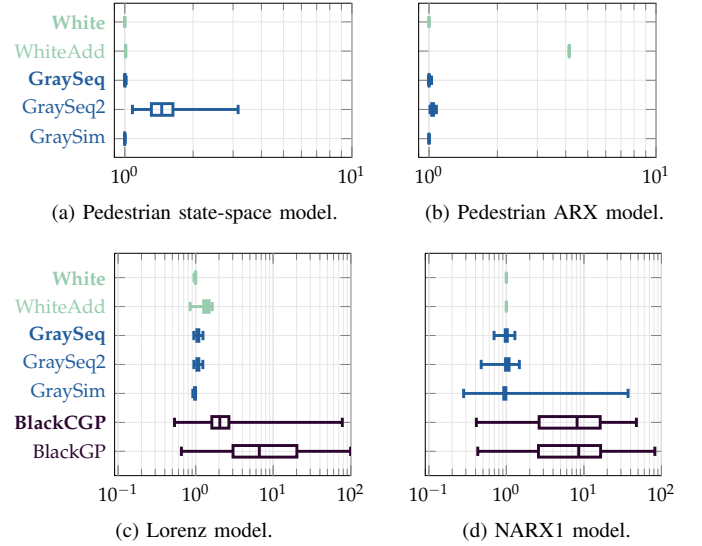


Fig. 3. Normalized cost for the identified models for different non-failure test suites.

for the computation of the reachable set, *White* identifies more conservative uncertainty sets for the nonlinear models, resulting in larger reachable sets after longer time horizons (see Fig. 4d at $k = n_p + 9$). The magnitude of the conservatism depends on the degree of the nonlinearities of the model.

The performance of *WhiteAdd* compared to *White* depends on how well additive uncertainties can approximate the uncertainties of the true systems. The true uncertainties of the pedestrian state-space model and the NARX1 model have a similar influence as additive uncertainties. As a result, the normalized cost of the models identified with *WhiteAdd* is almost identical to the cost of the models identified with *White* for these systems (see Figs. 3a and 3d). However, the true uncertainties of the pedestrian ARX model and Lorenz model have a more complex influence on the system, which cannot be well approximated by additive uncertainties. This can lead to bigger uncertainty sets and reachable sets (see Figs. 4b and 4c) and, thus, higher normalized costs (see Figs. 3b and 3c). In some cases, we also obtain models that are not reachset-conformant for validation test cases (see Fig. 4c, where the measurements are not contained in the reachable sets $\mathcal{Y}^{\text{WhiteAdd}}$ at $k = n_p + 9$).

If we know how the uncertainties act on the system, we should use this knowledge in the identification by applying the proposed algorithm *White*. The computation times are very short, and the identification results are as good as or better than those of an identification algorithm that only considers additive uncertainty sets.

2) **Gray-Box Identification**: We evaluate the gray-box identification approaches from Sec. V by augmenting the identification variables from the previous subsection with the model parameters p . Furthermore, we assume that the center vector of the uncertainty set of the NARX1 model is unknown, such that c_u is added to the identification variables for this model.

As we can see from Fig. 3, the proposed sequential approach, termed *GraySeq*, yields the smallest cost values for

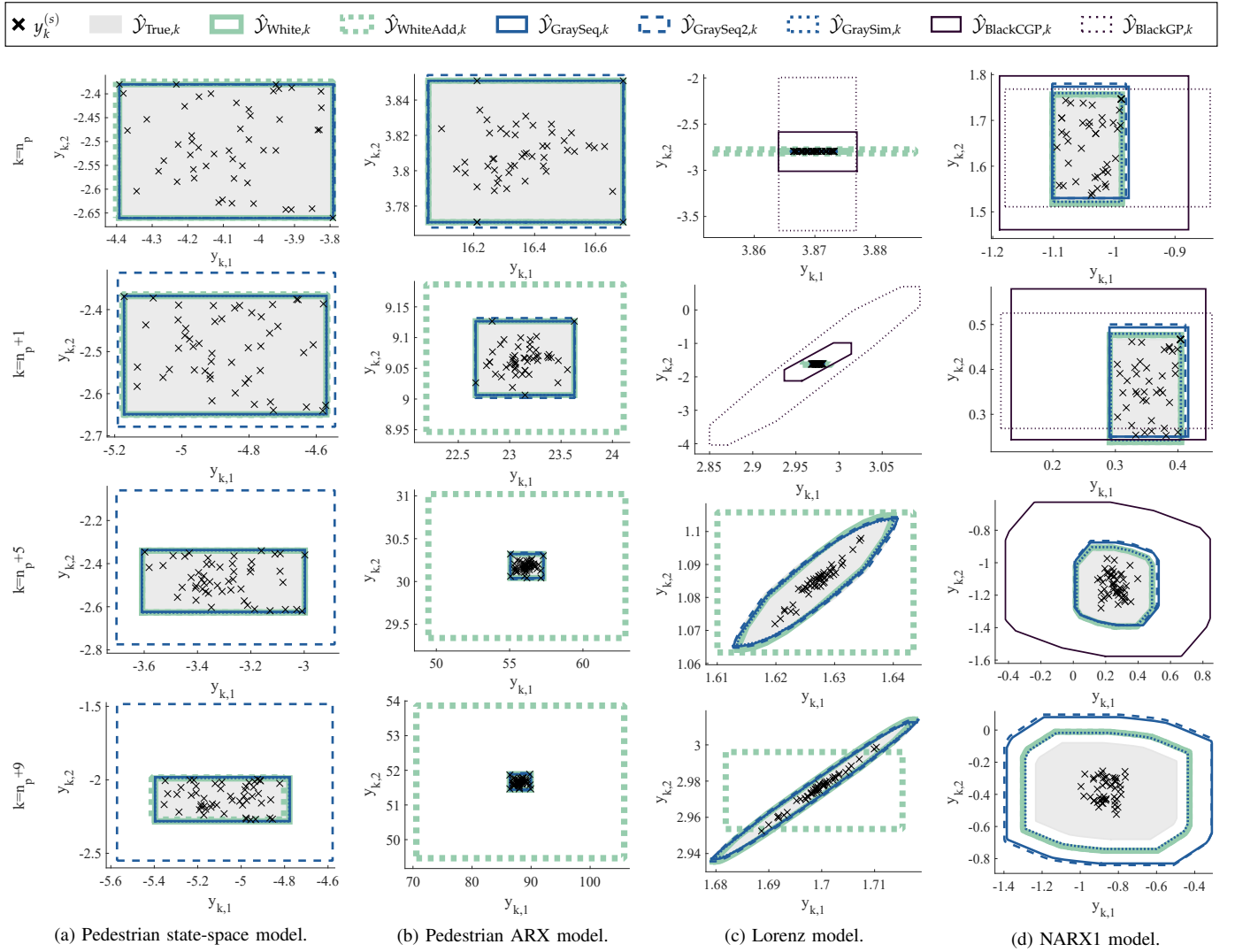


Fig. 4. Example test cases: measurements $y_k^{(s)}$ with $s = 1, \dots, 50$, reachable set $\mathcal{Y}_{\text{true},k}$ of the true system, and reachable sets $\mathcal{Y}_{\text{method},k}$ from models identified with different white, gray, and black-box identification methods for the time steps $k = n_p, n_p + 1, n_p + 5, n_p + 9$.

the linear models and the NARX1 model of all gray-box identification approaches, while the simultaneous identification approach *GraySim*, results in marginally smaller cost values than *GraySeq* for the Lorenz model. The baseline method *GraySeq2*, which uses a least-squares cost function, leads to the shortest computation times but also higher normalized costs and bigger reachable sets (see Table II and Figs. 3 and 4). As displayed in Table II, all three identification methods lead to small root mean squared errors between the estimated parameters and the true parameters, where *GraySeq* shows the smallest error on average. *GraySim* has the highest failure rate.

3) Black-Box Identification: Lastly, we evaluate the black-box identification approaches. As the model structure of linear models is completely determined by the linearity and the system dimension, which could be estimated by comparing the gray-box identification results for different dimensions, we only apply our black-box identification approaches to the two nonlinear systems. We assume no prior knowledge of the system state and thus identify the model function $\mathcal{F} = \{h\}$ of an input-output model, as described in (14).

First, we generate additional training datasets $\mathcal{M}_{T1}$ consisting of 1500 test cases for step 1 of Alg. 4 and $\mathcal{M}_{T2}$ consisting of 100 test cases for step 2. Each test case contains the measurements resulting from $n_s = 10$ uncertainty trajectories with a length of $n_k = n_p + 6$ time steps. For the model order, we choose $n_p = 3$ for approximating the Lorenz model and $n_p = 2$ for the NARX1 model. The accuracy of our black-box identification approach can be improved by using more complex models and more identification data. As such improvements always involve a trade-off in computation time, the decision will depend on the accuracy required for the application.

Next, we evolve a population of model functions with GP, using functionalities from the MATLAB toolbox GPTIPS2 [50]. The standard GP algorithm *BlackGP* aims to minimize the least-squares errors for the training data over 100 generations for a population of 300 randomly initialized model functions. From the final population, we choose the model function with the highest fit to the data. Additionally, we evaluate the CGP approach, denoted as *BlackCGP*, where

we minimize the least-squares cost function over $n_g = 95$ generations for $n_f = 300$ model functions, followed by $\tilde{n}_g = 5$ generations, where the cost ℓ_C of (22) is minimized for $\tilde{n}_f = 100$ model functions.

As the considered identification methods are of stochastic origin, the data generation and black-box identification process is repeated 30 times. The normalized cost for all test suites is displayed in the boxplots in Fig. 3 for the final models from all GP runs.

While both approaches lead to similar normalized costs for the NARX1 model (see Fig. 3d), the costs for identifying the Lorenz system with *BlackCGP* are much smaller (see Fig. 3c). This is also shown in Figs. 4c and 4d (we do not display the reachable sets $\mathcal{Y}_{\text{BlackCGP},k}$ and $\mathcal{Y}_{\text{BlackGP},k}$ at the time steps $k = n_p + 5$ and $k = n_p + 9$ in cases where they are too large): The models identified with *BlackCGP* mimic the reachability of the true system well in an overapproximative way. In contrast, the results of *BlackGP* are more volatile. The identified Lorenz model depicted in Fig. 4c leads to more conservative reachable sets, while the reachable set of the identified NARX model depicted in Fig. 4d is too small, as it does not contain all validation measurements at time step $k = n_p + 1$. Furthermore, as can be seen from Table II, the models identified with *BlackGP* lead more often to failures in the conformance identification step. These failures result partially from exploding reachable sets when considering set-based inputs and partially from an insufficient influence of the input on the model dynamics, i.e., by increasing the scaling factors of the input set, we are not able to enlarge the reachable set in all dimensions, especially at early time steps, such that not all measurements can be enclosed and the model cannot be made reachset-conformant.

C. Identification of a Vehicle Model

Finally, we test our identification method with measurements of the automated vehicle EDGAR [55], which is depicted in Fig. 1. Given the limited number of available test cases, we only validate our white-box and gray-box identification methods. We use the following kinematic model [56]:

$$\begin{aligned} \dot{x}_t &= \begin{bmatrix} x_{t,4} \cos(\beta_t + x_{t,5}) \\ x_{t,4} \sin(\beta_t + x_{t,5}) \\ r_\delta u_{t,1} \\ u_{t,2} \\ x_{t,4} \cos(\beta_t) \frac{\tan(x_{t,3})}{l} \end{bmatrix} + u_{t,3:7} \\ y_t &= \begin{bmatrix} x_{t,1} \\ x_{t,2} \\ \frac{1}{r_\delta} x_{t,3} \\ x_{t,4} \end{bmatrix} + u_{t,8:11}, \end{aligned}$$

where the slip angle β_t can be computed as

$$\beta_t = \arctan \left(\tan(x_{t,3}) \frac{l_r}{l} \right).$$

The state vector $x_t = [p_{x,t} \ p_{y,t} \ \delta_{f,t} \ v_t \ \psi_t]^\top$ consists of the position in x and y direction, the steering angle at the front axle, the vehicle velocity, and the yaw angle, respectively. We obtain noisy measurements of the position, $y_{t,1} = \hat{p}_{x,t}$ and

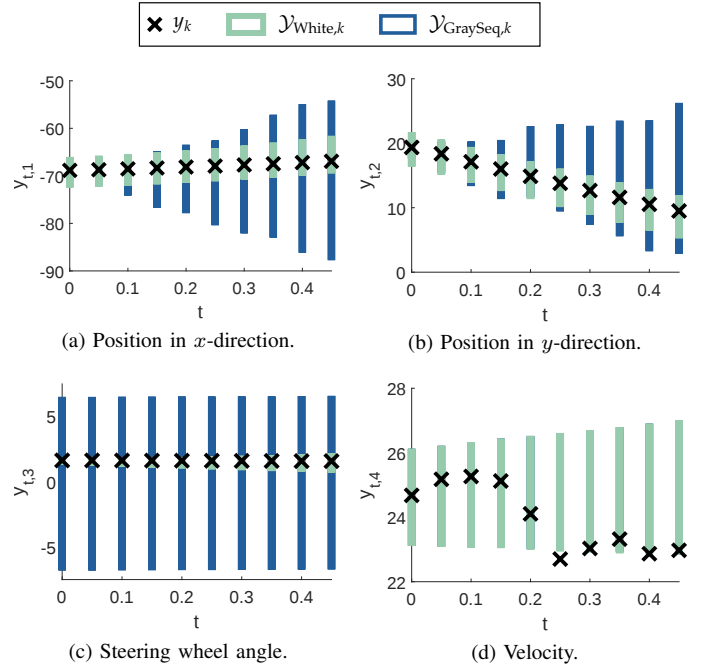


Fig. 5. Reachable sets and measurement trajectories for a failure test case of the automated vehicle EDGAR.

$y_{t,2} = \hat{p}_{y,t}$, the steering wheel angle $y_{t,3} = \hat{\delta}_{s,t}$, and the velocity $y_{t,4} = \hat{v}_t$. The input and disturbance vector $u_t = [\dot{\delta}_{s,t} \ a_{x,t} \ w_t^\top \ v_t^\top]^\top$ is composed of the steering wheel speed $\dot{\delta}_{s,t}$, the longitudinal acceleration $a_{x,t}$, the process noise vector w_t , and the measurement noise vector v_t . The parameters r_δ , l , and l_r denote the transmission ratio from δ_s to δ_f , the wheelbase, and the distance from the front axle to the center of gravity, respectively, and are given for the white-box identification and unknown for the gray-box identification.

The vehicle data is randomly split into an identification and a validation data set. The initial state estimate is computed from the measurements with $x_{*,0} = [\hat{p}_{x,0} \ \hat{p}_{y,0} \ r_\delta \hat{\delta}_{s,0} \ \hat{v}_0 \ \hat{\psi}_0]^\top$ where the initial yaw angle is calculated as

$$\hat{\psi}_0 = \arctan \left(\frac{\hat{p}_{y,1} - \hat{p}_{y,0}}{\hat{p}_{x,1} - \hat{p}_{x,0}} \right).$$

From the $n_m = 84$ test cases of the identification data set with a length of $n_k = 6$ time steps and $n_s = 1$, we identify the following uncertainty sets with the white-box identification approach from Alg. 1:

$$\begin{aligned} \mathcal{X}_0^{(m)} &= \langle \mathbf{0}, \text{diag}([0.00 \ 0.55 \ 0.00 \ 0.66 \ 0.00]) \rangle, \\ \mathcal{U}_{\text{in}} &= \langle \mathbf{0}, \text{diag}([0.00 \ 0.00]) \rangle, \\ \mathcal{U}_w &= \langle \mathbf{0}, \text{diag}([0.03 \ 0.02 \ 0.00 \ 0.06 \ 0.00]) \rangle, \\ \mathcal{U}_v &= \langle \mathbf{0}, \text{diag}([2.82 \ 1.84 \ 0.05 \ 0.79]) \rangle, \end{aligned}$$

with $\mathcal{U}_k^{(m)} = \mathcal{U}_{\text{in}} \times \mathcal{U}_w \times \mathcal{U}_v$. The large values in $\mathcal{U}_v$ indicate a large measurement uncertainty. The sequential gray-box identification approach *GraySeq* leads to larger initial state and process noise uncertainties.

Next, we validate these results by computing the reachable sets for the $n_m = 304$ test cases of the validation data set with a length of $n_k = 10$ time steps. 98.9% of the measurements are contained in their corresponding reachable set for the

white-box model and 93.6% for the gray-box model, which shows that the proposed identification framework works well for real-world data. One of the failure test cases is visualized in Fig. 5. We can see in Fig. 5d that the measurement at time 0.25s is marginally outside the reachable set (the reachable velocities are identical for both models for this test case). Introducing safety factors for scaling the identified uncertainty sets decreases the failure rate, i.e., a safety factor of $\epsilon = 1.2$ leads to 99.2% contained measurements for the white-box model and 97.6% for the gray-box model, while all measurements are contained in the respective reachable sets for both models when using a safety factor of $\epsilon = 3$. The magnitude of the necessary safety factor shows that the data contains some outliers (measurements that deviate substantially from the expected outputs). Thus, a lower failure rate could also be achieved by detecting and removing outliers from the measurement data.

VIII. CONCLUSIONS

We present a novel framework for identifying reachset-conformant models for linear and nonlinear system dynamics, adaptable to various levels of prior knowledge. A pivotal contribution is the introduction of the GLO approximation, which unifies reachability analysis across different model types. Leveraging this approximation, we introduce the first reachset-conformant identification method capable of handling potentially non-additive uncertainty sets and input-output models. Our framework also extends to the estimation of unknown model parameters and the identification of reachset-conformant black-box models using genetic programming, allowing for modeling complex systems with minimal prior information. Through extensive numerical experiments involving the identification of linear state-space, ARX, nonlinear state-space, and NARX models, as well as the modeling of an automated vehicle using real-world data, we demonstrate the robustness and practical applicability of our proposed approaches. Across all experiments, we successfully identify reachset-conformant models, whose reachable sets tightly enclose all measurements, within competitive computation times.

While our work guarantees reachset conformance for all trajectories considered in the identification process, we did not make any mathematical statements about the generalizability of the results to unseen data. Although our experiments demonstrate promising performance of the identified models on unseen data, future research will focus on deriving probabilistic guarantees, potentially utilizing the scenario approach [57]. Despite these limitations, our approach offers a valuable tool for enhancing reliability and mitigating conservatism in the formal verification and control of real-world systems. It represents a significant step towards transferring model-based verification results to practical applications, thereby addressing a critical gap in current methodologies.

REFERENCES

- [1] H. Roehm, J. Oehlerking, M. Woehle, and M. Althoff, "Model conformance for cyber-physical systems: A survey," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 3, 2019. article no. 30.
- [2] L. Ljung, *System Identification: Theory for the User*. Prentice Hall, 1999.
- [3] R. Isermann and M. Münchhof, *Identification of Dynamic Systems*. Springer, 2011.
- [4] K. J. Keesman, *System Identification: An Introduction*. Springer, 2011.
- [5] R. Tóth, *Modeling and Identification of Linear Parameter-Varying Systems*. Springer, 2010.
- [6] B. T. Polyak, S. A. Nazin, C. Durieu, and E. Walter, "Ellipsoidal parameter or state estimation under model uncertainty," *Automatica*, vol. 40, no. 7, pp. 1171–1179, 2004.
- [7] M. Canale, L. Fagiano, and M. C. Signorile, "Design of robust predictive control laws using set membership identified models," *Asian Journal of Control*, vol. 15, no. 6, pp. 1714–1722, 2013.
- [8] E. Walter and H. Piet-Lahanier, "Estimation of parameter bounds from bounded-error data: A survey," *Mathematics and Computers in Simulation*, vol. 32, no. 5-6, pp. 449–468, 1990.
- [9] M. Kieffer, E. Walter, and I. Simeonov, "Guaranteed nonlinear parameter estimation for continuous-time dynamical models," *IFAC Proceedings Volumes*, vol. 39, no. 1, pp. 843–848, 2006.
- [10] B. T. Polyak and S. A. Nazin, "Interval parameter estimation under model uncertainty," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 118–123, 2005.
- [11] M. Casini, A. Garulli, and A. Vicino, "Feasible parameter set approximation for linear models with bounded uncertain regressors," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 2910–2920, 2014.
- [12] J. Bravo, T. Alamo, and E. Camacho, "Bounded error identification of systems with time-varying parameters," *IEEE Transactions on Automatic Control*, vol. 51, no. 7, pp. 1144–1150, 2006.
- [13] A. Rauh, J. Kersten, and H. Aschemann, "Interval methods and contractor-based branch-and-bound procedures for verified parameter identification of quasi-linear cooperative system models," *Journal of Computational and Applied Mathematics*, vol. 367, pp. 1–10, 2019.
- [14] N. R. Mahato, L. Jaulin, S. Chakraverty, and J. Dezert, "Validated enclosure of uncertain nonlinear equations using SIVIA Monte Carlo," in *Recent Trends in Wave Mechanics and Vibrations*, pp. 455–468, Springer, 2020.
- [15] M. Kieffer and E. Walter, "Guaranteed estimation of the parameters of nonlinear continuous-time models: Contributions of interval analysis," *International Journal of Adaptive Control and Signal Processing*, vol. 25, no. 3, pp. 191–207, 2011.
- [16] S. H. Mo and J. P. Norton, "Fast and robust algorithm to compute exact polytope parameter bounds," *Mathematics and Computers in Simulation*, vol. 32, no. 5-6, pp. 481–493, 1990.
- [17] A. Alanwar, A. Koch, F. Allgöwer, and K. H. Johansson, "Data-driven reachability analysis using matrix zonotopes," in *Proc. of the 3rd Conference on Learning for Dynamics and Control*, vol. 144, pp. 163–175, 2021.
- [18] Y. Chen and N. Ozay, "Data-driven computation of robust control invariant sets with concurrent model selection," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 2, pp. 495–506, 2022.
- [19] S. Sadreddini and C. Belta, "Formal guarantees in data-driven model identification and control synthesis," in *Proc. of the 21st ACM International Conference on Hybrid Systems: Computation and Control*, pp. 147–156, 2018.
- [20] S. K. Mulagaleti, A. Bemporad, and M. Zanon, "Data-driven synthesis of robust invariant sets and controllers," *IEEE Control Systems Letters*, vol. 6, pp. 1676–1681, 2022.
- [21] A. El-Guindy, D. Han, and M. Althoff, "Formal analysis of drum-boiler units to maximize the load-following capabilities of power plants," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4691–4702, 2016.
- [22] B. Schürmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Köster, and M. Althoff, "Ensuring drivability of planned motions using formal methods," in *Proc. of the IEEE International Conference on Intelligent Transportation Systems*, pp. 1–8, 2017.
- [23] N. Kochdumper and S. Bak, "Conformant synthesis for Koopman operator linearized control systems," in *Proc. of the 61st IEEE Conference on Decision and Control*, pp. 7327–7332, 2022.
- [24] H. Roehm, A. Rausch, and M. Althoff, "Reachset conformance and automatic model adaptation for hybrid systems," *Mathematics*, vol. 10, no. 19, 2022. article no. 3567.
- [25] M. Althoff, A. Giusti, S. B. Liu, and A. Pereira, "Effortless creation of safe robots from modules through self-programming and self-verification," *Science Robotics*, vol. 4, no. 31, p. eaaw1924, 2019.
- [26] L. Schäfer, F. Gruber, and M. Althoff, "Scalable computation of robust control invariant sets of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 69, no. 2, pp. 755–770, 2024.

- [27] M. Althoff and J. M. Dolan, "Reachability computation of low-order models for the safety verification of high-order road vehicle models," in *Proc. of the American Control Conference*, pp. 3559–3566, 2012.
- [28] N. Kochdumper, A. Tarraf, M. Rechmal, M. Olbrich, L. Hedrich, and M. Althoff, "Establishing reachset conformance for the formal analysis of analog circuits," in *Proc. of the 25th Asia and South Pacific Design Automation Conference*, pp. 199–204, 2020.
- [29] F. Gruber and M. Althoff, "Scalable robust safety filter with unknown disturbance set," *IEEE Transactions on Automatic Control*, vol. 68, no. 12, pp. 7756–7770, 2023.
- [30] S. B. Liu, B. Schürmann, and M. Althoff, "Guarantees for real robotic systems: Unifying formal controller synthesis and reachset-conformant identification," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3776–3790, 2023.
- [31] M. Althoff, "Checking and establishing reachset conformance in CORA 2023," in *Proc. of the Workshop on Applied Verification of Continuous and Hybrid Systems*, pp. 9–33, 2023.
- [32] L. Lützwow and M. Althoff, "Scalable reachset-conformant identification of linear systems," *IEEE Control Systems Letters*, vol. 8, pp. 520–525, 2024.
- [33] L. Lützwow and M. Althoff, "Recursive identification of reachset-conformant models using constraint underapproximation," in *Proc. of the American Control Conference*, pp. 2530–2537, 2025.
- [34] A. Girard, "Reachability of uncertain linear systems using zonotopes," in *Proc. of the 8th ACM International Conference on Hybrid Systems: Computation and Control*, pp. 291–305, 2005.
- [35] W. Kühn, "Rigorously computed orbits of dynamical systems without the wrapping effect," *Computing*, vol. 61, no. 1, pp. 47–67, 1998.
- [36] M. Althoff, O. Stursberg, and M. Buss, "Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes," *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 233–249, 2010.
- [37] A. Platzer and E. M. Clarke, "The image computation problem in hybrid systems model checking," in *Hybrid Systems: Computation and Control*, pp. 473–486, 2007.
- [38] T. Gan, M. Chen, Y. Li, B. Xia, and N. Zhan, "Reachability analysis for solvable dynamical systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 2003–2018, 2018.
- [39] B. Finkbeiner, M. Fränzle, F. Kohn, and P. Kröger, "A truly robust signal temporal logic: Monitoring safety properties of interacting cyber-physical systems under uncertain observation," *Algorithms*, vol. 15, no. 4, 2022.
- [40] L. Jaulin, M. Kieffer, and O. Didrit, *Applied Interval Analysis*. Springer, 2006.
- [41] M. Althoff, "Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets," in *Proc. of the 16th ACM International Conference on Hybrid Systems: Computation and Control*, pp. 173–182, 2013.
- [42] M. Althoff and J. J. Rath, "Comparison of guaranteed state estimators for linear time-invariant systems," *Automatica*, vol. 130, 2021. article no. 109662.
- [43] L. Lützwow and M. Althoff, "Reachability analysis of ARMAX models," in *Proc. of the 62nd IEEE Conference on Decision and Control*, pp. 7021–7028, 2023.
- [44] S. B. Liu and M. Althoff, "Online verification of impact-force-limiting control for physical human-robot interaction," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 777–783, 2021.
- [45] N. Mohajerin and S. L. Waslander, "Multistep prediction of dynamic systems with recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3370–3383, 2019.
- [46] O. Nelles and R. Isermann, "Basis function networks for interpolation of local linear models," in *Proc. of 35th IEEE Conference on Decision and Control*, vol. 1, pp. 470–475 vol.1, 1996.
- [47] J. A. K. Suykens, B. L. R. De Moor, and J. Vandewalle, "Nonlinear system identification using neural state space models, applicable to robust control design," *International Journal of Control*, vol. 62, no. 1, pp. 129–152, 1995.
- [48] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsen, and A. Juditsky, "Nonlinear black-box modeling in system identification: A unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.
- [49] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," *Statistics and Computing*, vol. 4, no. 2, pp. 87–112, 1994.
- [50] D. P. Seanson, *GPTIPS 2: An Open-Source Software Platform for Symbolic Data Mining*, pp. 551–573. Springer, 2015.
- [51] M. Althoff, "An introduction to CORA 2015," in *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, pp. 120–151, 2015.
- [52] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," in *Proc. of the 47th IEEE Conference on Decision and Control*, pp. 4042–4048, 2008.
- [53] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.
- [54] A. Kroll and H. Schulte, "Benchmark problems for nonlinear system identification and control using soft computing methods: Need and overview," *Applied Soft Computing*, vol. 25, pp. 496–513, 2014.
- [55] P. Karle, T. Betz, M. Bosk, F. Fent, N. Gehrke, M. Geisslinger, L. Gressenbuch, P. Hafemann, S. Huber, M. Hübner, S. Huch, G. Kaljavesi, T. Kerbl, D. Kulmer, T. Mascetta, S. Maierhofer, F. Pfab, F. Rezabek, E. Rivera, S. Sagmeister, L. Seidlitz, F. Sauerbeck, I. Tahiraj, R. Trauth, N. Uhlemann, G. Würsching, B. Zarrouki, M. Althoff, J. Betz, K. Bengler, G. Carle, F. Diermeyer, J. Ott, and M. Lienkamp, "EDGAR: An autonomous driving research platform - from feature development to real-world application," *Preprint*, pp. 1–19, 2023. <https://arxiv.org/abs/2309.15492>.
- [56] R. Rajamani, *Lateral Vehicle Dynamics*. Springer, 2012.
- [57] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, 2006.



Laura Lützwow received the Bachelor of Science degree in mechatronics from the Technical University of Ilmenau, Germany, in 2020, and the Master of Science degree in robotics, cognition and intelligence from the Technical University of Munich, Germany, in 2022. In 2022, she was a visiting student researcher at the Massachusetts Institute of Technology, USA. Currently, she is working toward a Ph.D. degree in computer science with the Cyber-Physical Systems Group, School of Computation, Information and Technology, at the Technical University of Munich. Her research interests include system identification, control theory, and reachability analysis, with applications to safety-critical systems.



Matthias Althoff (Member, IEEE) received the Diploma Engineering degree in mechanical engineering and the Ph.D. degree in electrical engineering, both from the Technical University of Munich, Germany, in 2005 and 2010, respectively. From 2010 to 2012, he was a Postdoctoral Researcher at Carnegie Mellon University, Pittsburgh, PA, USA, and from 2012 to 2013, he was an Assistant Professor at the Technical University of Ilmenau, Germany. He is currently an Associate Professor in computer science with

the Technical University of Munich. His research interests include formal verification of continuous and hybrid systems, reachability analysis, planning algorithms, nonlinear control, robotics, automated vehicles, and power systems.