Fully Automated Verification of Linear Systems Using Inner- and Outer-Approximations of Reachable Sets

Mark Wetzlinger, Niklas Kochdumper, Stanley Bak, and Matthias Althoff

***Abstract*— Reachability analysis is a formal method to guarantee safety of dynamical systems under the influence of uncertainties. A substantial bottleneck of all reachability algorithms is the necessity to adequately tune specific algorithm parameters, such as the time step size, which requires expert knowledge. In this work, we solve this issue with a fully automated reachability algorithm that tunes all algorithm parameters internally such that the reachable set enclosure respects a user-defined approximation error bound in terms of the Hausdorff distance to the exact reachable set. Moreover, this bound can be used to extract an inner-approximation of the reachable set from the outer-approximation using the Minkowski difference. Finally, we propose a novel verification algorithm that automatically refines the accuracy of the outer-approximation and inner-approximation until specifications given by time-varying safe and unsafe sets can be verified or falsified. The numerical evaluation demonstrates that our verification algorithm successfully verifies or falsifies benchmarks from different domains without requiring manual tuning.**

***Index Terms*— Formal verification, reachability analysis, linear systems, set-based computing.**

I. Introduction

DEPLOYING cyber-physical systems in safety-critical environments requires formal verification techniques to ensure correctness with respect to the desired functionality, as failures can lead to severe economic or ecological consequences and loss of human life. One of the main techniques to provide safety guarantees is reachability analysis, which predicts all possible future system behaviors under uncertainty in the initial state and input. Reachability analysis has already been successfully applied in a wide variety of applications, such as analog/mixed-signal circuits [1], power systems [2], robotics [3], system biology [4], aerospace applications [5], and autonomous driving [6]. The most common verification tasks for these applications are reach-avoid problems, where one aims to prove that the system reaches a goal set while avoiding unsafe sets. A substantial bottleneck of all verification algorithms is the manual tuning of certain algorithm parameters, which requires expert knowledge. To overcome this limitation, we present the first fully automated verification algorithm for linear time-invariant systems.

This paragraph of the first footnote will contain the date on which you submitted your paper for review. This work was supported by the European Research Council (ERC) project justITSELF under grant agreement No 817629, by the German Research Foundation (DFG) project ConVeY under grant number GRK 2428, and by the Air Force Office of Scientific Research and the Office of Naval Research under award number FA9550-19-1-0288, FA9550-21-1-0121, FA9550-22-1-0450 and N00014-22-1-2156.

Mark Wetzlinger and Matthias Althoff are with the Department of Computer Science, Technical University of Munich, 85748 Garching, Germany (e-mail: {m.wetzlinger, althoff}@tum.de). Niklas Kochdumper and Stanley Bak are with the Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA (e-mail: {niklas.kochdumper, stanley.bak}@stonybrook.edu.

A. State of the Art

The exact reachable set can only be computed in rare special cases [7]. Therefore, one usually computes tight outer-approximations or inner-approximations of the reachable set instead, which can be used to either prove or disprove safety, respectively. While there exist many different approaches, e.g., stochastic techniques [8] or data-driven/learning methods [9], [10], the following review focuses on model-based reachability analysis of linear systems.

Most work is concerned with computing outer-approximations, where the most prominent approaches for linear systems are based on set propagation [11]–[13]. These methods evaluate the analytical solution for linear systems in a set-based manner and iteratively propagate the resulting reachable sets forward in time. One can propagate the sets from the previous step or the initial set. The latter method avoids the so-called wrapping effect [14], i.e., the amplification of outer-approximation errors over subsequent steps, but deals less efficiently with time-varying inputs. An alternative to set propagation is to compute the reachable set using widened trajectories from simulation runs [15], [16]. Moreover, special techniques have been developed recently to facilitate the analysis of high-dimensional systems. One strategy is to decompose the system into several decoupled/weakly-coupled blocks to reduce the computational effort [17], [18], while another group computes the reachable set in a lower-dimensional Krylov subspace that captures the dominant dynamical behavior [19], [20]. Inner-approximation algorithms have been researched for systems with piecewise-constant inputs based on set propagation [21], piecewise-affine systems based on linear matrix inequalities [22], and time-varying linear systems based on ellipsoidal inner-approximations to parametric integrals [23]. Other approaches compute inner-approximations by extraction from outer-approximations [24] or for projected dimensions [25]—despite being designed for nonlinear dynamics, these

works can still compete with the aforementioned specialized approaches for linear systems due to their recency.

Concerning the set representations for reachability analysis, early approaches used polyhedra or template polyhedra [26], which are limited to low-dimensional systems due to the exponential increase in the representation size; ellipsoids [23] were also used but result in a conservative approximation as they are not closed under Minkowski sum. More recent approaches use support functions [14], zonotopes [11], or their combination [27], for which all relevant set operations can be computed accurately and efficiently. A related representation is star sets [28], where constraints are imposed on a linear combination of base vectors.

Some approaches explicitly address the requirement of tightening the computed outer-approximation for successful verification. Many of them are based on counterexample-guided abstraction refinement (CEGAR), where either the model [1], [29] or the set representation [30], [31] is refined. A more recent approach [32] utilizes the relation between all algorithm parameters and the tightness of the reachable sets, proposing an individual parameter refinement in fixed discrete steps to yield tighter results. Another method [33] refines the tightness of the reachable set enclosure as much as the real-time constraints allow for re-computation in order to choose between a verified but conservative controller and an unverified counterpart with better performance.

Common reachability tools for linear systems are *CORA* [34], *Flow** [35], *HyDRA* [36], *HyLAA* [37], *JuliaReach* [38], *SpaceEx* [13], and *XSpeed* [39]. These tools still require manual tuning of algorithm parameters to obtain tight approximations. Since this requires expert knowledge about the underlying algorithms, the usage of reachability analysis is currently mainly limited to academia. Another issue is that the unknown distance between the computed outer-approximation and the exact reachable set, which may result in so-called spurious counterexamples. Recently, first steps towards automated parameter tuning have been taken: A rather brute-force method [40] proposes to recompute the reachable set from scratch, where the parameter values are refined using fixed scaling factors after each run. However, recomputation is a computationally demanding procedure and does not exploit information about the specific system dynamics. Another work [41] tunes the time step size by approximating the flow below a user-defined error bound but is limited to affine systems. The most sophisticated approach [13], [42] tunes the time step size by iterative refinement to eventually satisfy a user-defined error bound between the exact reachable set and the computed outer-approximation. Since this error bound is limited to manually selected directions, the obtained information may differ significantly depending on their choice. In conclusion, there does not yet exist a fully automated parameter tuning algorithm for linear systems that satisfies an error bound in terms of the Haussdorf distance to the exact reachable set.

B. Overview

This work is structured as follows: After introducing some preliminaries in Sec. II, we provide a mathematical formulation of the problem statement in Sec. III. The main body

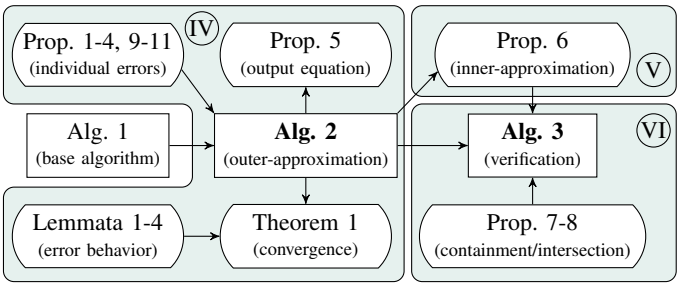


Fig. 1. Overview of theoretical contributions in Secs. IV-VI.

(Secs. IV-VI) builds upon our previous results [43], where we tuned the algorithm parameters based on non-rigorous approximation error bounds using a naive tuning strategy. Neither inner-approximations nor automated verification/falsification were considered in [43]. In detail, this article contributes the following novelties:

- We derive a rigorous approximation error for the reachable set, based on which we provide an *automated reachability algorithm* (Alg. 2) that adaptively tunes all algorithm parameters so that any desired error bound in terms of the Hausdorff distance between the exact reachable set and the computed outer-approximation is respected at all times (see Sec. IV).
- We show how to efficiently extract an *inner-approximation* from the previously computed outer-approximation (see Sec. V).
- We introduce an *automated verifier* (Alg. 3), which iteratively refines the accuracy of the outer- and inner-approximations until the specifications given by time-varying safe/unsafe sets can be either proven or disproven (see Sec. VI).

Fig. 1 depicts the relation of individual contributions in Secs. IV-VI. Finally, we demonstrate the performance of the proposed algorithms on a variety of numerical examples in Sec. VII and discuss directions for future work in Sec. VIII.

II. Preliminaries

A. Notation

Scalars and vectors are denoted by lowercase letters, matrices are denoted by uppercase letters. Given a vector $v \in \mathbb{R}^n$, $v_{(i)}$ represents the i-th entry and $\|v\|_p$ its p-norm. Similarly, for a matrix $M \in \mathbb{R}^{m \times n}$, $M_{(i,\cdot)}$ refers to the i-th row and $M_{(\cdot,j)}$ to the j-th column. The identity matrix of dimension n is denoted by I_n, the concatenation of two matrices M_1, M_2 by $[M_1 \ M_2]$, and we use $\mathbf{0}$ and $\mathbf{1}$ to represent vectors and matrices of proper dimension containing only zeros or ones, respectively. The operation $\mathrm{diag}(v)$ returns a square matrix with the vector v on its diagonal. Exact sets are denoted by standard calligraphic letters $\mathcal{S}$, outer-approximations by $\widehat{\mathcal{S}}$, and inner-approximations by $\check{\mathcal{S}}$. The empty set is represented by $\emptyset$. Moreover, we write v for the set $\{v\}$ consisting only of the point v. We refer to the radius of the smallest hypersphere centered at the origin and enclosing a set $\mathcal{S}$ by $\mathrm{rad}(\mathcal{S})$. The operation $\mathrm{box}(\mathcal{S})$ denotes the tightest axis-aligned interval outer-approximation of $\mathcal{S}$. Interval matrices are denoted by bold calligraphic letters: $\boldsymbol{\mathcal{M}} = [\underline{M}, \overline{M}] = \{M \in \mathbb{R}^{m \times n} \mid \underline{M} \leq$

$M \leq \overline{M}\}$, where the inequality is evaluated element-wise. Intervals are a special case of interval matrices, where the lower and upper bounds are vectors. Additionally, we define an n-dimensional hyperball centered at the origin by $\mathcal{B}_\varepsilon = \{x \in \mathbb{R}^n \mid \|x\|_2 \leq \varepsilon\} \subset \mathbb{R}^n$ with respect to the Euclidean norm. The floor operation $\lfloor x \rfloor$ returns the next smaller integer for a scalar x. For clarify, arguments of functions are sometimes omitted. Finally, we use $f(t) \sim \mathcal{O}(t^a)$ to represent that the function $f(t)$ approaches its limit value (0 or infinity in this work) as fast or faster than t^a.

B. Definitions

All sets are assumed to be compact, convex, and bounded. In this work, we represent outer-approximations of reachable sets with zonotopes [11, Def. 1]:

Definition 1 (Zonotope): *Given a center vector $c \in \mathbb{R}^n$ and a generator matrix $G \in \mathbb{R}^{n\times\gamma}$, a zonotope $\mathcal{Z} \subset \mathbb{R}^n$ is*

$$\mathcal{Z} := \left\{ c + \sum_{i=1}^{\gamma} G_{(\cdot,i)}\, \alpha_i \;\middle|\; \alpha_i \in [-1,1] \right\}.$$

The zonotope order is defined as $\rho := \frac{\gamma}{n}$ and we use the shorthand $\mathcal{Z} = \langle c, G \rangle_Z$.

Moreover, we represent inner-approximations of reachable sets with constrained zonotopes [44, Def. 3]:

Definition 2 (Constrained zonotope): *Given a vector $c \in \mathbb{R}^n$, a generator matrix $G \in \mathbb{R}^{n\times\gamma}$, a constraint matrix $A \in \mathbb{R}^{h\times\gamma}$, and a constraint offset $b \in \mathbb{R}^h$, a constrained zonotope $\mathcal{CZ} \subset \mathbb{R}^n$ is*

$$\mathcal{CZ} := \left\{ c + \sum_{i=1}^{\gamma} G_{(\cdot,i)}\, \alpha_i \;\middle|\; \sum_{i=1}^{\gamma} A_{(\cdot,i)} \alpha_i = b,\ \alpha_i \in [-1,1] \right\}.$$

We use the shorthand $\mathcal{CZ} = \langle c, G, A, b \rangle_{CZ}$.

Finally, unsafe sets and safe sets are represented by polytopes [45, Sec. 1.1]:

Definition 3 (Polytope): *Given a constraint matrix $C \in \mathbb{R}^{a\times n}$ and a constraint offset $d \in \mathbb{R}^a$, the halfspace representation of a polytope $\mathcal{P} \subset \mathbb{R}^n$ is*

$$\mathcal{P} := \{ x \in \mathbb{R}^n \mid Cx \leq d \}.$$

Equivalently, one can use the vertex representation

$$\mathcal{P} := \left\{ \sum_{i=1}^{s} \beta_i\, v_i \;\middle|\; \sum_{i=1}^{s} \beta_i = 1,\ \beta_i \geq 0 \right\},$$

where $\{v_1, \ldots, v_s\} \in \mathbb{R}^n$ are the polytope vertices. We use the shorthands $\mathcal{P} = \langle C, d \rangle_H$ and $\mathcal{P} = \langle [v_1 \ldots v_s] \rangle_V$.

Given the sets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^n, \mathcal{S}_3 \subset \mathbb{R}^m$ and a matrix $M \in \mathbb{R}^{w\times n}$, we require the set operations linear map $M\mathcal{S}_1$, Cartesian product $\mathcal{S}_1 \times \mathcal{S}_3$, Minkowski sum $\mathcal{S}_1 \oplus \mathcal{S}_2$, Minkowski difference $\mathcal{S}_1 \ominus \mathcal{S}_2$, and linear combination $\operatorname{comb}(\mathcal{S}_1, \mathcal{S}_2)$, which are defined as

$$M\mathcal{S}_1 = \{ Ms \mid s \in \mathcal{S}_1 \}, \tag{1}$$

$$\mathcal{S}_1 \times \mathcal{S}_3 = \{ [s_1^\top \ s_3^\top]^\top \mid s_1 \in \mathcal{S}_1, s_3 \in \mathcal{S}_3 \}, \tag{2}$$

$$\mathcal{S}_1 \oplus \mathcal{S}_2 = \{ s_1 + s_2 \mid s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2 \}, \tag{3}$$

$$\mathcal{S}_1 \ominus \mathcal{S}_2 = \{ s \mid s \oplus \mathcal{S}_2 \subseteq \mathcal{S}_1 \}, \tag{4}$$

$$\begin{aligned} \operatorname{comb}(\mathcal{S}_1, \mathcal{S}_2) = \{ &\lambda s_1 + (1-\lambda) s_2 \mid \\ &s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2, \lambda \in [0,1] \}. \end{aligned} \tag{5}$$

For zonotopes $\mathcal{Z}_1 = \langle c_1, G_1 \rangle_Z, \mathcal{Z}_2 = \langle c_2, G_2 \rangle_Z \subset \mathbb{R}^n$, linear map and Minkowski sum can be computed as [12, Eq. (2.1)]

$$M\mathcal{Z}_1 = \langle Mc_1, MG_1 \rangle_Z, \tag{6}$$

$$\mathcal{Z}_1 \oplus \mathcal{Z}_2 = \langle c_1 + c_2, [G_1 \ G_2] \rangle_Z. \tag{7}$$

Since the convex hull represents an enclosure of the linear combination, we furthermore obtain [12, Eq. (2.2)]

$$\begin{aligned} \operatorname{comb}(\mathcal{Z}_1, \mathcal{Z}_2) \subseteq \big\langle & 0.5(c_1 + c_2),\ \big[0.5(c_1 - c_2) \\ & 0.5(G_1 + G_2^{(1)})\ \ 0.5(G_1 - G_2^{(1)})\ \ G_2^{(2)} \big] \big\rangle_Z \end{aligned} \tag{8}$$

with

$$G_2^{(1)} = [G_{2(\cdot,1)} \ \ldots \ G_{2(\cdot,\gamma_1)}],\ G_2^{(2)} = [G_{2(\cdot,\gamma_1+1)} \ \ldots \ G_{2(\cdot,\gamma_2)}],$$

where we assume without loss of generality that $\mathcal{Z}_2$ has more generators than $\mathcal{Z}_1$ so that we can write the formula in a compact form. Later on in Sec. V, we show that the Minkowski difference of two zonotopes can be represented as a constrained zonotope. The multiplication $\boldsymbol{\mathcal{I}}\,\mathcal{Z}$ of an interval matrix $\boldsymbol{\mathcal{I}}$ with a zonotope $\mathcal{Z}$ can be outer-approximated as specified in [46, Thm. 4], and the operation $\operatorname{box}(\mathcal{Z})$ returns the axis-aligned box outer-approximation according to [12, Prop. 2.2]. The representation size of a zonotope can be decreased with zonotope order reduction. While our reachability algorithm is compatible with all common reduction techniques [47], we focus on Girard's method [11, Sec. 3.4] for simplicity:

Definition 4 (Zonotope order reduction): *Given a zonotope $\mathcal{Z} = \langle c, G \rangle_Z \subset \mathbb{R}^n$ and a desired zonotope order $\rho_d \geq 1$, the operation $\operatorname{reduce}(\mathcal{Z}, \rho_d) \supseteq \mathcal{Z}$ returns an enclosing zonotope with order smaller or equal to ρ_d:*

$$\operatorname{reduce}(\mathcal{Z}, \rho_d) = \langle c, [G_{keep} \ G_{red}] \rangle_Z,$$

with

$$G_{keep} = [G_{(\cdot,\pi_{\chi+1})} \ \ldots \ G_{(\cdot,\pi_\gamma)}],\ \ G_{red} = \operatorname{diag}\left(\sum_{i=1}^{\chi} |G_{(\cdot,\pi_i)}| \right),$$

where $\pi_1, \ldots, \pi_\gamma$ are the indices of the sorted generators

$$\|G_{(\cdot,\pi_1)}\|_1 - \|G_{(\cdot,\pi_1)}\|_\infty \leq \ldots \leq \|G_{(\cdot,\pi_\gamma)}\|_1 - \|G_{(\cdot,\pi_\gamma)}\|_\infty,$$

and $\chi = \gamma - \lfloor (\rho_d - 1)n \rfloor$ is the number of reduced generators.

For distances between sets, we use the Hausdorff distance:

Definition 5 (Hausdorff distance): *For two compact sets $\mathcal{S}_1, \mathcal{S}_2 \subseteq \mathbb{R}^n$, the Hausdorff distance with respect to the Euclidean norm is defined as*

$$\begin{aligned} d_H(\mathcal{S}_1, \mathcal{S}_2) = \max \Big\{ &\max_{s_1 \in \mathcal{S}_1} \Big(\min_{s_2 \in \mathcal{S}_2} \|s_1 - s_2\|_2 \Big), \\ &\max_{s_2 \in \mathcal{S}_2} \Big(\min_{s_1 \in \mathcal{S}_1} \|s_1 - s_2\|_2 \Big) \Big\}. \end{aligned} \tag{9}$$

Using a hyperball $\mathcal{B}_\varepsilon$ of radius ε, an alternative definition is

$$d_H(\mathcal{S}_1, \mathcal{S}_2) = \varepsilon \Leftrightarrow \mathcal{S}_2 \subseteq \mathcal{S}_1 \oplus \mathcal{B}_\varepsilon \wedge \mathcal{S}_1 \subseteq \mathcal{S}_2 \oplus \mathcal{B}_\varepsilon. \tag{10}$$

Moreover, we will frequently use the operator

$$\mathrm{err}(\mathcal{S}) := \mathrm{rad}(\mathrm{box}(\mathcal{S})). \tag{11}$$

As an immediate consequence of (11), we obtain

$$d_H\big(\mathbf{0}, \mathcal{S}\big) \overset{\mathbf{0}\in\mathcal{S}}{\leq} \mathrm{err}(\mathcal{S}), \tag{12}$$

which states that the Hausdorff distance between a set and the origin can be bounded by the radius of an enclosing hyperball.

III. Problem Formulation

We consider linear time-invariant systems of the form

$$\dot{x}(t) = Ax(t) + Bu(t) + p, \tag{13}$$
$$y(t) = Cx(t) + Wv(t) + q, \tag{14}$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{\ell\times n}$, $W \in \mathbb{R}^{\ell\times o}$, where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the input, $y(t) \in \mathbb{R}^\ell$ is the output, $v(t) \in \mathbb{R}^o$ is a measurement error, $p \in \mathbb{R}^n$ is a constant input, and $q \in \mathbb{R}^\ell$ is a constant offset on the output. The initial state $x(t_0)$ is uncertain within the initial set $\mathcal{X}^0 \subset \mathbb{R}^n$, the input $u(t)$ is uncertain within the input set $\mathcal{U} \subset \mathbb{R}^m$, and $v(t)$ is uncertain within the set of measurement errors $\mathcal{V} \subset \mathbb{R}^o$. In this work, we assume that $\mathcal{X}^0, \mathcal{U}$, and $\mathcal{V}$ are represented by zonotopes. Using $\mathcal{U} = \langle c_u, G_u\rangle_Z$, we define the vector $\tilde{u} = Bc_u + p \in \mathbb{R}^n$ and the set $\mathcal{U}_0 = \langle \mathbf{0}, BG_u\rangle_Z \subset \mathbb{R}^n$ for later derivations. Please note that we assume a constant vector $\tilde{u}$ to keep the presentation simple, but the extension to time-varying inputs $\tilde{u}(t)$ is straightforward. Without loss of generality, we set the initial time to $t_0 = 0$ and the time horizon to $[0, t_{\text{end}}]$. The reachable set is defined as follows:

Definition 6 (Reachable set): *Let us denote the solution to* (13) *for the initial state* $x(0)$ *and the input signal* $u(\cdot)$ *by* $\xi(t; x(0), u(\cdot))$. *Given an initial set* $\mathcal{X}^0$ *and an input set* $\mathcal{U}$, *the reachable set at time* $t \geq 0$ *is*

$$\mathcal{R}(t) := \big\{\xi(t; x(0), u(\cdot)) \;\big|\; x(0) \in \mathcal{X}^0, \forall\theta \in [0, t] \colon u(\theta) \in \mathcal{U}\big\}.$$

We denote the time-point reachable set at time $t = t_k$ *by* $\mathcal{R}(t_k)$ *and the time-interval reachable set over* $\tau_k \in [t_k, t_{k+1}]$ *by* $\mathcal{R}(\tau_k) := \bigcup_{t\in[t_k,t_{k+1}]} \mathcal{R}(t)$.

Since the exact reachable set as defined in Def. 6 cannot be computed for general linear systems [7], we aim to compute tight outer-approximations $\widehat{\mathcal{R}}(t) \supseteq \mathcal{R}(t)$ and inner-approximations $\check{\mathcal{R}}(t) \subseteq \mathcal{R}(t)$ instead.

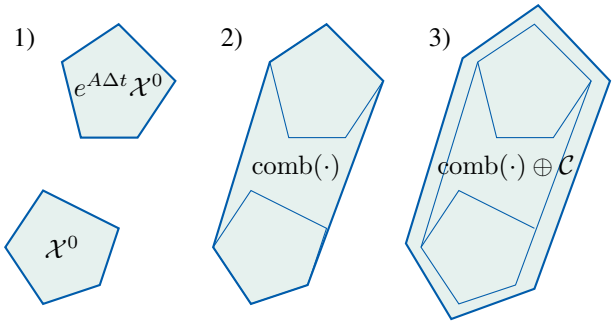


Fig. 2. Visualization of the three steps required to compute the homogeneous solution of the first time interval, adapted from [12, Fig. 3.1].

Algorithm 1 Reachability algorithm (manual tuning)

Require: Linear system $\dot{x} = Ax + Bu + p$, initial set $\mathcal{X}^0 = \langle c_x, G_x\rangle_Z$, input set $\mathcal{U} = \langle c_u, G_u\rangle_Z$, time horizon t_{end}, time step size Δt dividing the time horizon into an integer number of steps, truncation order η, zonotope order ρ
Ensure: Outer-approximation of the reachable set $\widehat{\mathcal{R}}([0, t_{\text{end}}])$

1: $t_0 \leftarrow 0, \tilde{u} \leftarrow Bc_u + p, \mathcal{U}_0 \leftarrow \langle \mathbf{0}, BG_u\rangle_Z$
2: $\mathcal{H}(t_0) \leftarrow \mathcal{X}^0, \mathcal{P}^u(t_0) \leftarrow \mathbf{0}, \widehat{\mathcal{P}}^{\mathcal{U}}(t_0) \leftarrow \mathbf{0}$
3: $\mathcal{P}^u(\Delta t) \leftarrow$ Eq. (19), $\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t) \leftarrow$ Eq. (20)
4: **for** $k \leftarrow 0$ to $\frac{t_{\text{end}}}{\Delta t} - 1$ **do**
5: $\quad t_{k+1} \leftarrow t_k + \Delta t, \tau_{k+1} \leftarrow [t_k, t_{k+1}]$
6: $\quad \mathcal{P}^u(t_{k+1}) \leftarrow \mathcal{P}^u(t_k) \oplus e^{At_k}\mathcal{P}^u(\Delta t)$
7: $\quad \mathcal{H}(t_{k+1}) \leftarrow e^{At_{k+1}}\mathcal{X}^0 + \mathcal{P}^u(t_{k+1})$
8: $\quad \mathcal{C} \leftarrow \boldsymbol{\mathcal{F}}(\Delta t, \eta)\mathcal{H}(t_k) \oplus \boldsymbol{\mathcal{G}}(\Delta t, \eta)\tilde{u}$ ▷ see (15)-(16)
9: $\quad \widehat{\mathcal{H}}(\tau_k) \leftarrow \mathrm{comb}\big(\mathcal{H}(t_k), \mathcal{H}(t_{k+1})\big) \oplus \mathcal{C}$
10: $\quad \widehat{\mathcal{P}}^{\mathcal{U}}(t_{k+1}) \leftarrow \mathrm{reduce}\big(\widehat{\mathcal{P}}^{\mathcal{U}}(t_k) \oplus e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t), \rho\big)$
11: $\quad \widehat{\mathcal{R}}(\tau_k) \leftarrow \widehat{\mathcal{H}}(\tau_k) \oplus \widehat{\mathcal{P}}^{\mathcal{U}}(t_{k+1})$
12: **end for**
13: $\widehat{\mathcal{R}}([0, t_{\text{end}}]) \leftarrow \bigcup_{k=0}^{\frac{t_{\text{end}}}{\Delta t}-1} \widehat{\mathcal{R}}(\tau_k)$

Our automated tuning approach is based on Alg. 1, which is a slight modification of the wrapping-free propagation-based reachability algorithm [14]. Fundamentally, one exploits the superposition principle of linear systems by separately computing the homogeneous and particular solutions, which are then combined in Line 11 to yield the overall reachable set for each time interval. The homogeneous solution can be enclosed using the following three steps visualized in Fig. 2:

1) Compute the time-point solution by propagating the initial set with the exponential matrix $e^{A\Delta t}$.
2) Approximate the time-interval solution with the linear combination (8), which would only enclose straight-line trajectories.
3) Account for the curvature of the trajectories by enlarging the linear combination with the set $\mathcal{C}$.

The curvature enclosure $\mathcal{C}$ is computed in Line 8 of Alg. 1 using the interval matrices [12, Sec. 3.2]

$$\boldsymbol{\mathcal{F}}(\Delta t, \eta) = \bigoplus_{i=2}^{\eta} \mathcal{I}_i(\Delta t)\,\frac{A^i}{i!} \oplus \boldsymbol{\mathcal{E}}(\Delta t, \eta) \tag{15}$$

$$\boldsymbol{\mathcal{G}}(\Delta t, \eta) = \bigoplus_{i=2}^{\eta+1} \mathcal{I}_i(\Delta t)\,\frac{A^{i-1}}{i!} \oplus \boldsymbol{\mathcal{E}}(\Delta t, \eta)\Delta t \tag{16}$$

$$\text{with}\quad \mathcal{I}_i(\Delta t) = \big[\big(i^{\frac{-i}{i-1}} - i^{\frac{-1}{i-1}}\big)\Delta t^i, 0\big], \tag{17}$$

where the interval matrix $\boldsymbol{\mathcal{E}}(\Delta t_k, \eta_k)$ represents the remainder of the exponential matrix [12, Eq. (3.2)]:

$$\boldsymbol{\mathcal{E}}(\Delta t, \eta) = [-E(\Delta t, \eta), E(\Delta t, \eta)],$$
$$E(\Delta t, \eta) = e^{|A|\Delta t} - \sum_{i=0}^{\eta} \frac{\big(|A|\Delta t\big)^i}{i!}. \tag{18}$$

To increase the tightness, the particular solution $\mathcal{P}^u(\Delta t)$ due

to the constant input $\tilde{u}$ [12, Eq. (3.7)],

$$\mathcal{P}^u(\Delta t) = A^{-1}(e^{A\Delta t} - I_n)\,\tilde{u}, \tag{19}$$

is added to the homogeneous solution in Line 7. If the matrix A is not invertible, we can integrate A^{-1} in the power series of the exponential matrix to compute $\mathcal{P}^u(\Delta t)$. The particular solution due to the time-varying input within the set $\mathcal{U}_0$ can be enclosed by [12, Eq. (3.7)]

$$\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t) = \bigoplus_{i=0}^{\eta} \frac{A^i \Delta t^{i+1}}{(i+1)!}\,\mathcal{U}_0 \oplus \boldsymbol{\mathcal{E}}(\Delta t, \eta)\Delta t\,\mathcal{U}_0. \tag{20}$$

Finally, the reachable set $\widehat{\mathcal{R}}([0, t_{\text{end}}])$ for the entire time horizon is given by the union of the sets for individual time-interval reachable sets according to Line 13.

As for all other state-of-the-art reachability algorithms [11]–[14], [21], [43], the main disadvantage of Alg. 1 is that the tightness of the computed reachable set $\widehat{\mathcal{R}}(t)$ is unknown and heavily depends on the chosen time step size Δt, truncation order η, and zonotope order ρ. In this work, we solve both issues by automatically tuning these algorithm parameters such that the Hausdorff distance between the computed enclosure $\widehat{\mathcal{R}}(t)$ and the exact reachable set $\mathcal{R}(t)$ remains below a desired threshold $\varepsilon_{\max}$ at all times:

$$\text{Tune } \Delta t, \eta, \rho \quad \text{s.t.} \quad \forall t \in [0, t_{\text{end}}] : d_H\big(\mathcal{R}(t), \widehat{\mathcal{R}}(t)\big) \leq \varepsilon_{\max}.$$

We will further utilize this result to efficiently extract an inner-approximation of the reachable set (Sec. V) and construct a fully automated verification algorithm (Sec. VI).

IV. Automated Parameter Tuning

Let us now present our approach for the automated tuning of algorithm parameters. While Alg. 1 uses fixed values for Δt, η, and ρ, we tune different values Δt_k, η_k, and ρ_k in each step k based on the induced outer-approximation error in order to satisfy the error bound at all times. To achieve this, we first derive closed-form expressions describing how the individual errors depend on the values of each parameter in Sec. IV-A. Next, we present our automated parameter tuning algorithm in Sec. IV-B and prove its convergence in Sec. IV-C. Finally, we discuss further improvements to the algorithm in Sec. IV-D and describe the extension to output sets in Sec. IV-E.

A. Error Measures

Several sources of outer-approximation errors exist in Alg. 1:

1) **Affine dynamics (Line 9)**: The time-interval solution $\widehat{\mathcal{H}}(\tau_k)$ of the affine dynamics contains errors originating from enclosing the linear combination by zonotopes and the set $\mathcal{C}$ accounting for the curvature of trajectories.
2) **Particular solution (Lines 10-11)**: Using the outer-approximation $\widehat{\mathcal{P}}^{\mathcal{U}}(t_k)$ based on (20) for the particular solution due to the input set $\mathcal{U}_0$ induces another error. Moreover, the Minkowski addition of $\widehat{\mathcal{P}}^{\mathcal{U}}(t_{k+1})$ to $\widehat{\mathcal{H}}(\tau_k)$ is outer-approximative as it ignores dependencies in time.
3) **Zonotope order reduction (Line 10)**: The representation size of the particular solution $\widehat{\mathcal{P}}^{\mathcal{U}}(t_k)$ has to be reduced, which induces another error.

In this subsection, we derive upper bounds for all these errors in terms of the Hausdorff distance between an exact set $\mathcal{S}$ and the computed outer-approximation $\widehat{\mathcal{S}}$. We will use two different albeit related error notations: New errors induced in step k are denoted by $\Delta\varepsilon_k^*(\Delta t_k, \eta_k)$ and $\Delta\varepsilon_k^*(\rho_k)$ to emphasize the dependence on the respective algorithm parameters (using $*$ as a placeholder for various superscripts). Some errors for single time steps add up over time. We accumulate all previous errors $\varepsilon_k^*(\cdot)$ until time t_k by

$$\varepsilon_{k+1}^* = \varepsilon_k^* + \Delta\varepsilon_k^*(\cdot), \quad \varepsilon_0^* := 0. \tag{21}$$

Let us now derive closed-form expressions for all errors.

1) Affine Dynamics: We start by determining the error contained in the computed outer-approximation $\widehat{\mathcal{H}}(\tau_k)$ of the time-interval solution for the affine dynamics $\dot{x}(t) = Ax(t) + \tilde{u}$:

Proposition 1 (Affine dynamics error): *Given the set $\mathcal{H}(t_k) = \langle c_h, G_h \rangle_Z$ with $G_h \in \mathbb{R}^{n\times\gamma_h}$, the Hausdorff distance between the exact time-interval solution of the affine dynamics*

$$\mathcal{H}(\tau_k) = \big\{ e^{At}x(t_k) + A^{-1}(e^{A(t-t_k)} - I_n)\tilde{u} \;\big|\; t \in \tau_k,\, x(t_k) \in \mathcal{H}(t_k) \big\}$$

and the corresponding outer-approximation

$$\widehat{\mathcal{H}}(\tau_k) = \operatorname{comb}\big(\mathcal{H}(t_k), \mathcal{H}(t_{k+1})\big) \oplus \mathcal{C} \tag{22}$$

in Line 9 of Alg. 1 is bounded by

$$d_H\big(\mathcal{H}(\tau_k), \widehat{\mathcal{H}}(\tau_k)\big) \leq \Delta\varepsilon_k^h(\Delta t_k, \eta_k) := 2\operatorname{err}(\mathcal{C}) + \sqrt{\gamma_h}\,\big\|G_h^{(-)}\big\|_2, \tag{23}$$

where $G_h^{(-)} = (e^{A\Delta t_k} - I_n)G_h$.

Proof. An inner-approximation of $\mathcal{H}(\tau_k)$ is given by

$$\check{\mathcal{H}}(\tau_k) = \operatorname{comb}\big(\mathcal{H}(t_k), \mathcal{H}(t_{k+1})\big) \ominus \mathcal{B}_\mu \ominus \mathcal{C} \subseteq \mathcal{H}(\tau_k),$$

where we additionally have to subtract a hyperball of radius $\mu = \sqrt{\gamma_h}\,\big\|G_h^{(-)}\big\|_2$ bounding the Hausdorff distance between the exact linear combination (5) and the zonotope enclosure (8) according to Prop. 9 in Appendix A. The error $\Delta\varepsilon_k^{\text{h}}(\Delta t_k, \eta_k)$ bounds the distance between $\check{\mathcal{H}}(\tau_k)$ and the outer-approximation $\widehat{\mathcal{H}}(\tau_k)$ in (22). □

2) Particular Solution: We first account for the error induced by enclosing the exact time-point solution $\mathcal{P}^{\mathcal{U}}(t_{k+1})$ with the outer-approximation $\widehat{\mathcal{P}}^{\mathcal{U}}(t_{k+1})$:

Proposition 2 (Time-point error in particular solution): *The Hausdorff distance between the exact particular solution*

$$\mathcal{P}^{\mathcal{U}}(t_{k+1}) = \left\{ \int_0^{t_{k+1}} e^{A(t_{k+1}-\theta)}u(\theta)\,\mathrm{d}\theta \;\middle|\; u(\theta) \in \mathcal{U}_0 \right\}$$

and the recursively computed outer-approximation

$$\widehat{\mathcal{P}}^{\mathcal{U}}(t_{k+1}) = \widehat{\mathcal{P}}^{\mathcal{U}}(t_k) \oplus e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)$$

with $\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)$ computed according to (20) *is bounded by*

$$\varepsilon_{k+1}^{\mathcal{U}} = \varepsilon_k^{\mathcal{U}} + \Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k, \eta_k), \tag{24}$$

where the error $\Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k, \eta_k)$ for one time step is bounded

by

$$\begin{aligned}\Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k,\eta_k)&\\ :=\operatorname{err}&\left(e^{At_k}\left(\Big(\sum_{i=1}^{\eta_k}\tilde{A}_i\Big)\mathcal{U}_0\oplus\boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k)\Delta t_k\,\mathcal{U}_0\right)\right)\\ +\operatorname{err}&\left(e^{At_k}\left(\bigoplus_{i=1}^{\eta_k}\tilde{A}_i\,\mathcal{U}_0\oplus\boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k)\Delta t_k\,\mathcal{U}_0\right)\right)\end{aligned}\tag{25}$$

with $\tilde{A}_i=\frac{A^i\Delta t_k^{i+1}}{(i+1)!}$ *and* $\boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k)$ *from* (18).

Proof. The error $\Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k,\eta_k)$ for one time step is given by the Hausdorff distance between the computed outer-approximation and an inner-approximation obtained by considering constant inputs according to Prop. 11 in Appendix A. The overall error $\varepsilon_{k+1}^{\mathcal{U}}$ follows by error propagation as in (21). □

Since $0\in\mathcal{U}_0$, the added set due to uncertain inputs $e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\theta)$ is equal to $\mathbf{0}$ at the beginning of each time step ($\theta=0$) and monotonically grows towards the set $e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)$ at the end of the time step ($\theta=\Delta t_k$) as shown in Fig. 3 on the left. The Minkowski sum in Line 9 of Alg. 1 ignores this dependency on time, inducing another outer-approximation error:

Proposition 3 (Time-interval error in particular solution): *The maximum Hausdorff distance at any time* $t\in\tau_k=[t_k,t_{k+1}]$ *between the exact time-interval particular solution*

$$\mathcal{P}^{\mathcal{U}}(t)=\left\{\int_0^t e^{A(t-\theta)}u(\theta)\mathrm{d}\theta\;\middle|\;u(\theta)\in\mathcal{U}_0\right\}$$

and the outer-approximation $\forall t\in\tau_k:\mathcal{P}^{\mathcal{U}}(t)\subseteq\widehat{\mathcal{P}}^{\mathcal{U}}(t_{k+1})$ *is bounded by*

$$\max_{t\in[t_k,t_{k+1}]}d_H\big(\mathcal{P}^{\mathcal{U}}(t),\widehat{\mathcal{P}}^{\mathcal{U}}(t_{k+1})\big)\le\varepsilon_k^{\mathcal{U}}+\Delta\varepsilon_k^{\mathcal{U},\tau}(\Delta t_k,\eta_k)$$

with $\varepsilon_k^{\mathcal{U}}$ *from Prop. 2 and the additional error*

$$\Delta\varepsilon_k^{\mathcal{U},\tau}(\Delta t_k,\eta_k):=\operatorname{err}\left(e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)\right).\tag{26}$$

Proof. Since $e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\theta)$ grows monotonically with θ, the maximum deviation over the time interval τ_k occurs at $t=t_k$, where the actual additional set would be $\mathbf{0}$, but instead $e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)$ is used. Therefore, the error is

$$d_H\big(\mathbf{0},e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)\big)\overset{(12)}{\le}\operatorname{err}\left(e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)\right),$$

which corresponds to the size of the additional set. □

3) Zonotope Order Reduction: The zonotope order reduction of the particular solution $\widehat{\mathcal{P}}^{\mathcal{U}}(t_{k+1})$ in Line 10 of Alg. 1 induces another error. To determine this reduction error, we first split the particular solution $e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)$ into two parts

$$\begin{aligned}e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)&\overset{(20)}{=}e^{At_k}\left(\bigoplus_{i=0}^{\eta}\tilde{A}_i\,\mathcal{U}_0\oplus\boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k)\Delta t_k\,\mathcal{U}_0\right)\\ &=e^{At_k}\underbrace{\Delta t_k\,\mathcal{U}_0}_{=:\widehat{\mathcal{P}}_0^{\mathcal{U}}(\Delta t_k)}\oplus e^{At_k}\underbrace{\left(\bigoplus_{i=1}^{\eta}\tilde{A}_i\,\mathcal{U}_0\oplus\boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k)\Delta t\,\mathcal{U}_0\right)}_{=:\widehat{\mathcal{P}}_\infty^{\mathcal{U}}(\Delta t_k)}\end{aligned}\tag{27}$$

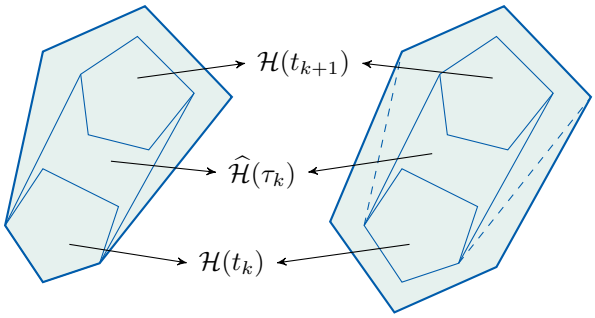


Fig. 3. Reachable set computation using the correct particular time-interval solution $\widehat{\mathcal{P}}^{\mathcal{U}}(\tau_k)$ (left) and an outer-approximation $\widehat{\mathcal{P}}^{\mathcal{U}}(t_{k+1})\supseteq\widehat{\mathcal{P}}^{\mathcal{U}}(\tau_k)$ (right).

with $\tilde{A}_i$ defined as in Prop. 2. We exploit that the error $\Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k,\eta_k)$ in (25) is unaffected by using the box outer-approximation $\operatorname{box}(e^{At_k}\widehat{\mathcal{P}}_\infty^{\mathcal{U}}(\Delta t_k))$ instead of $e^{At_k}\widehat{\mathcal{P}}_\infty^{\mathcal{U}}(\Delta t_k)$ since the box outer-approximation is also used in the computation of $\Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k,\eta_k)$. Therefore, we can always reduce $\widehat{\mathcal{P}}_\infty^{\mathcal{U}}(t_{k+1})$ to a box (which has zonotope order 1) as that reduction error is already contained in $\Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k,\eta_k)$. Consequently, we only have to determine the reduction error of $\widehat{\mathcal{P}}_0^{\mathcal{U}}(t_{k+1})$:

Proposition 4 (Zonotope order reduction error): *The Hausdorff distance between the particular solution* $\widehat{\mathcal{P}}_0^{\mathcal{U}}(t_{k+1})$ *computed without any reduction and its iteratively reduced counterpart* $\operatorname{reduce}\big(\widehat{\mathcal{P}}_0^{\mathcal{U}}(t_{k+1}),\rho_k\big)$ *is bounded by*

$$\varepsilon_{k+1}^{\mathrm{r}}=\varepsilon_k^{\mathrm{r}}+\Delta\varepsilon_k^{\mathrm{r}}(\rho_k),\tag{28}$$

where the error $\Delta\varepsilon_k^{\mathrm{r}}(\rho_k)$ *for one time step is bounded by*

$$\begin{aligned}&d_H\big(\widehat{\mathcal{P}}_0^{\mathcal{U}}(t_{k+1}),\operatorname{reduce}\big(\widehat{\mathcal{P}}_0^{\mathcal{U}}(t_{k+1}),\rho_k\big)\big)\\ &\le\Delta\varepsilon_k^{\mathrm{r}}(\rho_k):=\operatorname{err}(\langle\mathbf{0},G_{red}\rangle_Z),\end{aligned}\tag{29}$$

where G_{red} *defined as in Def. 4 contains the generators selected for reduction.*

Proof. The error $\Delta\varepsilon_k^{\mathrm{r}}(\rho_k)$ for one time step is given by the box enclosure of the zonotope formed by the generators selected for reduction. Using the error propagation formula (21), we then obtain the overall error $\varepsilon_{k+1}^{\mathrm{r}}$ in (28). □

4) Summary: The derived error terms allow us to compute an upper bound for the outer-approximation error contained in the time-point solution and time-interval solution:

$$d_H\big(\mathcal{R}(t_k),\widehat{\mathcal{R}}(t_k)\big)\le\varepsilon_k^{\mathcal{U}}+\varepsilon_k^{\mathrm{r}},\tag{30}$$

$$\begin{aligned}d_H\big(\mathcal{R}(\tau_k),\widehat{\mathcal{R}}(\tau_k)\big)&\le\varepsilon_k^{x}\\ :=\Delta\varepsilon_k^{\mathrm{h}}(\Delta t_k,\eta_k)&+\varepsilon_k^{\mathcal{U}}+\Delta\varepsilon_k^{\mathcal{U},\tau}(\Delta t_k,\eta_k)+\varepsilon_{k+1}^{\mathrm{r}},\end{aligned}\tag{31}$$

where we use $\varepsilon_k^{\mathcal{U}}$ instead of $\varepsilon_{k+1}^{\mathcal{U}}$ in (31), since the difference $\varepsilon_{k+1}^{\mathcal{U}}-\varepsilon_k^{\mathcal{U}}\overset{(24)}{=}\Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k,\eta_k)$ is already included in $\Delta\varepsilon_k^{\mathcal{U},\tau}(\Delta t_k,\eta_k)$. As usually only time-interval reachable sets are required for formal verification, we will only use the time-interval error ε_k^{x} in our automated parameter tuning algorithm.

B. Automated Tuning Algorithm

Using the error terms derived in the previous subsection, we now present an algorithm that tunes Δt_k, η_k, and ρ_k automatically such that the Hausdorff distance between the

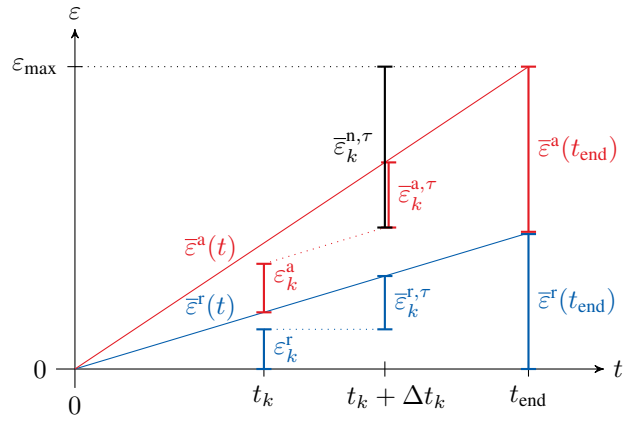


Fig. 4. The errors $\varepsilon^{\mathrm{a}}_{\boldsymbol{k}}$ and $\varepsilon^{\mathrm{r}}_{\boldsymbol{k}}$ until $\boldsymbol{t_k}$ and the bounds $\overline{\varepsilon}^{\mathrm{a}}(\boldsymbol{t})$ and $\overline{\varepsilon}^{\mathrm{r}}(\boldsymbol{t})$ yield the individual error bounds $\overline{\varepsilon}^{\mathrm{n},\boldsymbol{\tau}}_{\boldsymbol{k}}, \overline{\varepsilon}^{\mathrm{a},\boldsymbol{\tau}}_{\boldsymbol{k}}, \overline{\varepsilon}^{\mathrm{r},\boldsymbol{\tau}}_{\boldsymbol{k}}$ for the current step $\boldsymbol{k}$.

exact reachable set $\mathcal{R}([0, t_{\mathrm{end}}])$ and the computed enclosure $\widehat{\mathcal{R}}([0, t_{\mathrm{end}}])$ is below the error bound $\varepsilon_{\max}$. As different types of errors require different strategies for parameter tuning, we divide the derived errors into three categories:

1) **Non-accumulating error** $\Delta\varepsilon^{\mathrm{n}}_k(\Delta t_k, \eta_k)$: Since the errors $\Delta\varepsilon^{\mathrm{h}}_k(\Delta t_k, \eta_k)$ and $\Delta\varepsilon^{\mathcal{U},\tau}_k(\Delta t_k, \eta_k)$ only affect the current step, we define the non-accumulating error by

$$\Delta\varepsilon^{\mathrm{n}}_k(\Delta t_k, \eta_k) := \Delta\varepsilon^{\mathrm{h}}_k(\Delta t_k, \eta_k) + \Delta\varepsilon^{\mathcal{U},\tau}_k(\Delta t_k, \eta_k). \tag{32}$$

2) **Accumulating error** $\varepsilon^{\mathrm{a}}_k$: The particular solution $\widehat{\mathcal{P}}^{\mathcal{U}}(t_k)$ accumulates over time, yielding

$$\varepsilon^{\mathrm{a}}_k := \varepsilon^{\mathcal{U}}_k, \qquad \Delta\varepsilon^{\mathrm{a}}_k(\Delta t_k, \eta_k) := \Delta\varepsilon^{\mathcal{U}}_k(\Delta t_k, \eta_k), \tag{33}$$

for the overall accumulating error and the accumulating error for one time step.

3) **Reduction error** $\varepsilon^{\mathrm{r}}_k$: The representation size of the particular solution $\widehat{\mathcal{P}}^{\mathcal{U}}(t_k)$ is iteratively reduced (Line 10), which induces an accumulating error (28). Despite its accumulation, we do not add this error to $\varepsilon^{\mathrm{a}}_k$ since it does not directly depend on the time step size Δt_k.

We have to manage these errors over time so that the resulting set $\widehat{\mathcal{R}}(t)$ respects the error bound $\varepsilon_{\max}$ at all times. Therefore, we partition $\varepsilon_{\max}$ into individual admissible errors $\overline{\varepsilon}^{\mathrm{n},\tau}_k, \overline{\varepsilon}^{\mathrm{a},\tau}_k$, and $\overline{\varepsilon}^{\mathrm{r},\tau}_k$ for each step, which is visualized in Fig. 4:

1) **Reduction error bound** $\overline{\varepsilon}^{\mathrm{r},\tau}_k$: We limit the reduction error by a linearly increasing bound

$$\overline{\varepsilon}^{\mathrm{r}}(t) = \frac{t}{t_{\mathrm{end}}}\zeta\varepsilon_{\max}, \qquad \zeta \in [0, 1). \tag{34}$$

Thus, the additional error $\Delta\varepsilon^{\mathrm{r}}_k(\rho_k)$ in step k is bounded by

$$\overline{\varepsilon}^{\mathrm{r},\tau}_k = \overline{\varepsilon}^{\mathrm{r}}(t_k + \Delta t_k) - \varepsilon^{\mathrm{r}}_k, \tag{35}$$

i.e., the difference between the bound at time $t_k + \Delta t_k$ and the accumulated error until t_k. While our algorithm works for arbitrary values ζ, we present a heuristic for choosing ζ later in Sec. IV-D.

2) **Accumulating error bound** $\overline{\varepsilon}^{\mathrm{a},\tau}_k$: Similarly, we limit the accumulating error by another linearly increasing bound

$$\overline{\varepsilon}^{\mathrm{a}}(t) = \frac{t}{t_{\mathrm{end}}}(1 - \zeta)\varepsilon_{\max}, \tag{36}$$

so that we have $\overline{\varepsilon}^{\mathrm{r}}(t_{\mathrm{end}}) + \overline{\varepsilon}^{\mathrm{a}}(t_{\mathrm{end}}) = \varepsilon_{\max}$. Analogously to (35), the bound for the additional error $\Delta\varepsilon^{\mathcal{U}}_k(\Delta t_k, \eta_k)$ is

$$\overline{\varepsilon}^{\mathrm{a},\tau}_k = \overline{\varepsilon}^{\mathrm{a}}(t_k + \Delta t_k) - \varepsilon^{\mathrm{a}}_k. \tag{37}$$

3) **Non-accumulating error bound** $\overline{\varepsilon}^{\mathrm{n},\tau}_k$: Finally, we obtain the bound for the non-accumulating error $\Delta\varepsilon^{\mathrm{n}}_k(\Delta t_k, \eta_k)$ by subtracting the other two bounds from $\varepsilon_{\max}$:

$$\overline{\varepsilon}^{\mathrm{n},\tau}_k = \varepsilon_{\max} - \overline{\varepsilon}^{\mathrm{r}}(t_k + \Delta t_k) - \varepsilon^{\mathrm{a}}_k. \tag{38}$$

Note that we only subtract $\varepsilon^{\mathrm{a}}_k$ instead of $\overline{\varepsilon}^{\mathrm{a}}(t_k + \Delta t_k)$ for the accumulating error since the accumulating error $\Delta\varepsilon^{\mathrm{a}}_k(\Delta t_k, \eta_k) = \Delta\varepsilon^{\mathcal{U}}_k(\Delta t_k, \eta_k)$ for the current step is already accounted for by the error $\Delta\varepsilon^{\mathcal{U},\tau}_k(\Delta t_k, \eta_k)$, which is according to (32) part of the non-accumulating error. This also guarantees us a non-zero bound for $\overline{\varepsilon}^{\mathrm{n},\tau}_k$ in the last step even though $\overline{\varepsilon}^{\mathrm{a}}(t_{\mathrm{end}}) + \overline{\varepsilon}^{\mathrm{r}}(t_{\mathrm{end}}) = \varepsilon_{\max}$.

The tuning strategies for the parameters are as follows:

- **Time step size** Δt_k: We initialize Δt_k by its previous value Δt_{k-1}, or by t_{end} as an initial guess for the first step. To keep the presentation simple, we iteratively halve this value until the error bounds are satisfied; a more sophisticated tuning method is described in Sec. IV-D.
- **Truncation order** η_k: We tune η_k simultaneously with the computation of $\boldsymbol{\mathcal{F}}(\Delta t_k, \eta_k)$ and $\boldsymbol{\mathcal{G}}(\Delta t_k, \eta_k)$, for which the idea proposed in [48, Sec. 3.1] is reused: The partial sums

$$\boldsymbol{\mathcal{T}}^{(j)} = \bigoplus_{i=1}^{j} \mathcal{I}_i \frac{A^i}{i!} \tag{39}$$

in the computation of $\boldsymbol{\mathcal{F}}(\Delta t_k, \eta_k)$ in (15) are successively compared until the relative change in the Frobenius norm of $\boldsymbol{\mathcal{T}}^{(j)}$ computed according to [49, Thm. 10] is smaller than 10^{-10}. As this bound is relative, we can ensure convergence independently of the scale of the system, since the size of the additional terms decreases exponentially for $i \to \infty$.
- **Zonotope order** ρ_k: We iteratively increase the order ρ_k until the error $\Delta\varepsilon^{\mathrm{r}}_k(\rho_k)$ is smaller than the error bound $\overline{\varepsilon}^{\mathrm{r},\tau}_k$. A more efficient method compared to this naive implementation is to directly integrate the search for a suitable order into the zonotope order reduction.

The resulting automated tuning algorithm is shown in Alg. 2: In the repeat-until loop (Lines 6-15), we first decrease the time step size Δt_k (Line 7) and tune the truncation order η_k (Lines 9-12) until the respective error bounds $\overline{\varepsilon}^{\mathrm{a},\tau}_k$ and $\overline{\varepsilon}^{\mathrm{n},\tau}_k$ for the accumulating and non-accumulating errors are safisfied. After this loop, we compute the particular solution due to the input set $\mathcal{U}_0$ and tune the zonotope order ρ_k (Lines 20-22) yielding the reduction error $\Delta\varepsilon^{\mathrm{r}}_k(\rho_k)$. Afterwards, we compute the solution to the affine dynamics (Lines 25-28) and finally obtain the reachable set of the current time interval (Line 30).

The runtime complexity of Alg. 2 is $\mathcal{O}(n^3)$ as for the base algorithm, Alg. 1. For an initial set $\mathcal{X}^0$ with zonotope order ρ_X and an input set with zonotope order ρ_U, the space complexity for the k-th set $\widehat{\mathcal{R}}(\tau_k)$ is bounded by $\mathcal{O}(n^2(\rho_X + k\rho_U))$ and the space complexity for Alg. 2 then follows by summing over all individual steps.

Algorithm 2 Reachability algorithm (automated tuning)

Require: Linear system $\dot{x} = Ax + Bu + p$, initial set $\mathcal{X}^0 = \langle c_x, G_x \rangle_Z$, input set $\mathcal{U} = \langle c_u, G_u \rangle_Z$, time horizon t_{end}, error bound $\varepsilon_{\max}$

Ensure: Outer-approximation of the reachable set $\widehat{\mathcal{R}}([0, t_{\text{end}}])$

1: $k \leftarrow 0, t_0 \leftarrow 0, \Delta t_{-1} \leftarrow t_{\text{end}}, \mathcal{H}(t_0) \leftarrow \mathcal{X}^0$
2: $\mathcal{P}^u(t_0) \leftarrow \langle \mathbf{0}, [] \rangle_Z, \widehat{\mathcal{P}}_0^{\mathcal{U}}(t_0) \leftarrow \langle \mathbf{0}, [] \rangle_Z, \widehat{\mathcal{P}}_\infty^{\mathcal{U}}(t_0) \leftarrow \langle \mathbf{0}, [] \rangle_Z$
3: $\tilde{u} \leftarrow Bc_u + p, \mathcal{U}_0 \leftarrow \langle \mathbf{0}, BG_u \rangle_Z$
4: **while** $t_k < t_{\text{end}}$ **do**
5: $\quad \Delta t_k \leftarrow 2\Delta t_{k-1}$
6: $\quad$ **repeat**
7: $\quad\quad \Delta t_k \leftarrow \frac{1}{2}\Delta t_k,\ t_{k+1} \leftarrow t_k + \Delta t_k$
8: $\quad\quad \eta_k \leftarrow 0, \boldsymbol{\mathcal{T}}^{(\eta_k)} = \mathbf{0}$
9: $\quad\quad$ **repeat**
10: $\quad\quad\quad \eta_k \leftarrow \eta_k + 1$
11: $\quad\quad\quad \boldsymbol{\mathcal{T}}^{(\eta_k)} \leftarrow \boldsymbol{\mathcal{T}}^{(\eta_k - 1)} \oplus \mathcal{I}_{\eta_k} \frac{A^{\eta_k}}{\eta_k!}$ ▷ see (17), (39)
12: $\quad\quad$ **until** $1 - \left\|\boldsymbol{\mathcal{T}}^{(\eta_k-1)}\right\|_F / \left\|\boldsymbol{\mathcal{T}}^{(\eta_k)}\right\|_F \leq 10^{-10}$
13: $\quad\quad \Delta\varepsilon_k^{\text{n}}(\Delta t_k, \eta_k), \Delta\varepsilon_k^{\text{a}}(\Delta t_k, \eta_k) \leftarrow$ (32), (33)
14: $\quad\quad \bar{\varepsilon}_k^{\text{a},\tau}, \bar{\varepsilon}_k^{\text{n},\tau} \leftarrow$ (37), (38)
15: $\quad$ **until** $\Delta\varepsilon_k^{\text{a}}(\Delta t_k, \eta_k) \leq \bar{\varepsilon}_k^{\text{a},\tau} \wedge \Delta\varepsilon_k^{\text{n}}(\Delta t_k, \eta_k) \leq \bar{\varepsilon}_k^{\text{n},\tau}$
16: $\quad \widehat{\mathcal{P}}_0^{\mathcal{U}}(\Delta t_k), \widehat{\mathcal{P}}_\infty^{\mathcal{U}}(\Delta t_k) \leftarrow$ (27)
17: $\quad \widehat{\mathcal{P}}_\infty^{\mathcal{U}}(t_{k+1}) \leftarrow \widehat{\mathcal{P}}_\infty^{\mathcal{U}}(t_k) \oplus \text{box}\left(e^{At_k} \widehat{\mathcal{P}}_\infty^{\mathcal{U}}(\Delta t_k)\right)$
18: $\quad \widehat{\mathcal{P}}_0^{\mathcal{U}}(t_{k+1}) \leftarrow \widehat{\mathcal{P}}_0^{\mathcal{U}}(t_k) \oplus e^{At_k} \widehat{\mathcal{P}}_0^{\mathcal{U}}(\Delta t_k)$
19: $\quad \bar{\varepsilon}_k^{\text{r},\tau} \leftarrow$ (35), $\rho_k \leftarrow 0$
20: $\quad$ **repeat**
21: $\quad\quad \rho_k \leftarrow \rho_k + 1,\ \Delta\varepsilon_k^{\text{r}}(\rho_k) \leftarrow$ Prop. 4
22: $\quad$ **until** $\Delta\varepsilon_k^{\text{r}}(\rho_k) \leq \bar{\varepsilon}_k^{\text{r},\tau}$
23: $\quad \widehat{\mathcal{P}}^{\mathcal{U}}(t_{k+1}) \leftarrow \text{reduce}\left(\widehat{\mathcal{P}}_0^{\mathcal{U}}(t_{k+1}), \rho_k\right) \oplus \widehat{\mathcal{P}}_\infty^{\mathcal{U}}(t_{k+1})$
24: $\quad \mathcal{P}^u(\Delta t_k) \leftarrow$ (19)
25: $\quad \mathcal{P}^u(t_{k+1}) \leftarrow \mathcal{P}^u(t_k) \oplus e^{At_k} \mathcal{P}^u(\Delta t_k)$
26: $\quad \mathcal{H}(t_{k+1}) \leftarrow e^{At_{k+1}} \mathcal{X}^0 + \mathcal{P}^u(t_{k+1})$
27: $\quad \mathcal{C} \leftarrow \boldsymbol{\mathcal{F}}(\Delta t_k, \eta_k) \widehat{\mathcal{H}}(t_k) \oplus \boldsymbol{\mathcal{G}}(\Delta t_k, \eta_k) \tilde{u}$ ▷ see (15), (16)
28: $\quad \widehat{\mathcal{H}}(\tau_k) \leftarrow \text{comb}\left(\mathcal{H}(t_k), \mathcal{H}(t_{k+1})\right) \oplus \mathcal{C}$
29: $\quad \varepsilon_{k+1}^{\text{a}}, \varepsilon_{k+1}^{\text{r}} \leftarrow$ (33), (28)
30: $\quad \widehat{\mathcal{R}}(\tau_k) \leftarrow \widehat{\mathcal{H}}(\tau_k) \oplus \widehat{\mathcal{P}}^{\mathcal{U}}(t_{k+1})$
31: $\quad k \leftarrow k + 1$
32: **end while**
33: **return** $\widehat{\mathcal{R}}([0, t_{\text{end}}]) \leftarrow \bigcup_{j=0}^{k-1} \widehat{\mathcal{R}}(\tau_j)$

C. Proof of Convergence

While Alg. 2 guarantees to return a reachable set $\widehat{\mathcal{R}}([0, t_{\text{end}}])$ satisfying the error bound $\varepsilon_{\max}$ by construction, it remains to show that the algorithm terminates in finite time. To respect the linearly increasing bound for the accumulating error, we have to show that this error decreases faster than linearly with the time step size Δt_k; thus, by successively halving the time step size, we will always find a time step size so that the error bound is satisfied. Using Lemmas 1-4 from Appendix B, we now formulate our main theorem:

Theorem 1 (Convergence): *Alg. 2 terminates in finite time for arbitrary error bounds $\varepsilon_{max} > 0$.*

Proof. By Lemma 2, the additional accumulating error $\Delta\varepsilon_k^{\text{a}}(\Delta t_k, \eta_k)$ decreases quadratically with Δt_k. Thus, we are guaranteed to find a time step size that satisfies the linearly decreasing bound $\bar{\varepsilon}_k^{\text{a},\tau}$ by successively halving Δt_k. The non-accumulating error $\Delta\varepsilon_k^{\text{n}}(\Delta t_k, \eta_k)$ decreases at least linearly with Δt_k according to Lemmas 3-4. Since the error bound $\bar{\varepsilon}_k^{\text{n},\tau}$ approaches a constant value greater than 0 for $\Delta t_k \to 0$, we are therefore always able to safisfy $\bar{\varepsilon}_k^{\text{n},\tau}$ by reducing the time step size. The additional reduction error $\Delta\varepsilon_k^{\text{r}}(\rho_k)$ can be set to 0 by simply omitting the reduction, which trivially satisfies any bound $\bar{\varepsilon}_k^{\text{r},\tau}$. □

Our adaptive algorithm Alg. 2 must be based on a wrapping-free reachability algorithm to guarantee convergence as successive propagation with $e^{A\Delta t_k}$ would eliminate the required faster-than-linear decrease of the accumulating error.

D. Improved Tuning Methods

While Alg. 2 is guaranteed to converge, there is still room for improvement regarding the computation time. Hence, we present enhanced methods for adapting the time step size Δt and the choice of ζ in (34) determining the amount of error that is allocated for reduction.

1) Time Step Size: Ideally, the chosen time step size Δt_k fulfills the resulting error bounds as tightly as possible. To this end, we replace the naive adaptation of Δt_k in Line 7 of Alg. 2 by regression: We use the previously obtained error values as data points to define linear and quadratic approximation functions modeling the behavior of the error over Δt_k, depending on the asymptotic behavior of the respective errors according to Lemmas 1-4 from Appendix B. We then compute an estimate of the time step size required to satisfy the error bounds based on the approximation functions. This estimate is then refined until the error bounds are satisfied.

2) Reduction Error Allocation: The second major improvement is to pre-compute a near-optimal value for the parameter ζ in (34) using a heuristic that aims to minimize the zonotope order of the resulting reachable sets. Our heuristic is based on the following observation: For increasing values of ζ, more margin is allocated to the reduction error and less margin to the accumulating and non-accumulating errors. Thus, the total number of steps increases because the algorithm has to select smaller time step sizes, yielding a higher zonotope order. At the same time, the zonotope order can be lowered more due to the larger reduction error margin. We now want to determine the optimal value of ζ balancing these two effects.

We first estimate the zonotope order of $\widehat{\mathcal{P}}^{\mathcal{U}}(t_{\text{end}})$ using the number of time steps if reduction is completely omitted. Let us denote the total number of steps for Alg. 2 without reduction ($\zeta = 0$) by k_0', and the zonotope order of the input set $\mathcal{U}_0$ by $\rho_{\mathcal{U}}$. In each step, the set $e^{At_k} \widehat{\mathcal{P}}_0^{\mathcal{U}}(\Delta t_k) = e^{At_k} \Delta t_k \mathcal{U}_0$ is added to $\widehat{\mathcal{P}}^{\mathcal{U}}(t_k)$, which iteratively increases the zonotope of $\widehat{\mathcal{P}}^{\mathcal{U}}(t_k)$ by $\rho_{\mathcal{U}}$. Due to the linear decrease of the total error (see Lemmas 2-4 in Appendix B), using the value $\zeta = 0.5$ at most doubles the number of steps compared to $\zeta = 0$. Therefore,

the zonotope order of $\widehat{\mathcal{P}}^{\mathcal{U}}(t_{\text{end}}) = \widehat{\mathcal{P}}_0^{\mathcal{U}}(t_{\text{end}}) \oplus \widehat{\mathcal{P}}_\infty^{\mathcal{U}}(t_{\text{end}})$ can be estimated as

$$\rho^+(\zeta) = \frac{1}{1-\zeta} k_0' \rho_{\mathcal{U}} + 1, \tag{40}$$

if no order reduction takes place, where the summands represent the orders of $\widehat{\mathcal{P}}_0^{\mathcal{U}}(t_{\text{end}})$ and $\widehat{\mathcal{P}}_\infty^{\mathcal{U}}(t_{\text{end}})$, respectively.

Next, we estimate the zonotope order of $\widehat{\mathcal{P}}^{\mathcal{U}}(t_{\text{end}})$ for a non-zero value $\zeta > 0$ yielding a non-zero error margin $\bar{\varepsilon}^{\text{r}}(t) > 0$ that is used to reduce the order of $\widehat{\mathcal{P}}^{\mathcal{U}}(t_{\text{end}})$. Using a fixed time step size Δt yielding an integer number $k' = \frac{t_{\text{end}}}{\Delta t}$ of time steps, we compute the sequence

$$\forall j \in \{1, ..., k'+1\} : \tilde{\varepsilon}_j^{\text{r}} = \text{err}\left(e^{A(j-1)\Delta t} \Delta t \mathcal{U}_0\right),$$

which estimates the maximum reduction error in each time step. Let us introduce the ordering π which permutes $\{1, ..., k'+1\}$ such that $\tilde{\varepsilon}_{\pi_1}^{\text{r}} < ... < \tilde{\varepsilon}_{\pi_{k'+1}}^{\text{r}}$. To mimic the accumulation of the reduction error, we introduce the cumulative sum over all $\tilde{\varepsilon}_j^{\text{r}}$ ordered by π:

$$\forall j \in \{1, ..., k'+1\} : \sigma_j = \sum_{i=1}^{j+1} \tilde{\varepsilon}_{\pi_i}^{\text{r}}.$$

The maximum reducible order $\rho^-(\zeta)$ exploits the reduction error bound $\bar{\varepsilon}^{\text{r}}(t_{\text{end}}) = \zeta \varepsilon_{\max}$ as much as possible:

$$\begin{aligned} &\rho^-(\zeta) = j^* \rho_{\mathcal{U}}, \\ &\text{where } j^* = \underset{j \in \{1,...,k'+1\}}{\arg\max} \ \sigma_j \leq \zeta \varepsilon_{\max}. \end{aligned} \tag{41}$$

Finally, we combine the two parts (40) and (41) describing the counteracting influences to obtain the following heuristic:

$$\zeta = \underset{\zeta \in [0,1)}{\arg\min} \, \rho^+(\zeta) - \rho^-(\zeta). \tag{42}$$

Since this is a scalar optimization problem, we use a fine grid of different values for $\zeta \in [0, 1)$ to estimate the optimal value.

E. Extension to Output Sets

We now show how to extend the proposed algorithm to outputs $y(t)$. The output set $\mathcal{Y}(t)$ can be computed by evaluating the output equation (14) in a set-based manner:

$$\mathcal{Y}(t) = C\mathcal{R}(t) \oplus W\mathcal{V} + q. \tag{43}$$

For the outer-approximation error of the output set, we have to account for the linear transformation with the matrix C:

Proposition 5: *Consider a linear system of the form* (13)-(14). *Given the error* ε_k^x (31) *of the reachable set* $\widehat{\mathcal{R}}(\tau_k)$, *the corresponding output set* $\widehat{\mathcal{Y}}(\tau_k)$ *has an error of*

$$d_H\big(\mathcal{Y}(\tau_k), \widehat{\mathcal{Y}}(\tau_k)\big) \leq \varepsilon_k^y := \varepsilon_k^x \big\|C\big\|_2. \tag{44}$$

Proof. For the error in the state $x(t)$, we have

$$d_H\big(\mathcal{R}(\tau_k), \widehat{\mathcal{R}}(\tau_k)\big) \leq \varepsilon_k^x \overset{(10)}{\Rightarrow} \widehat{\mathcal{R}}(\tau_k) \subseteq \mathcal{R}(\tau_k) \oplus \mathcal{B}_\varepsilon,$$

where the hyperball $\mathcal{B}_\varepsilon$ has radius $\varepsilon = \varepsilon_k^x$. Applying the output equation (14) to the right-hand side yields

$$\begin{aligned} &C\widehat{\mathcal{R}}(\tau_k) \oplus W\mathcal{V} + q \subseteq C\mathcal{R}(\tau_k) \oplus C\mathcal{B}_\varepsilon \oplus W\mathcal{V} + q \\ \overset{(43)}{\Leftrightarrow} \quad &\widehat{\mathcal{Y}}(\tau_k) \subseteq \mathcal{Y}(\tau_k) \oplus C\mathcal{B}_\varepsilon. \end{aligned}$$

The error in $\widehat{\mathcal{Y}}(\tau_k)$ is therefore given by the radius of the smallest sphere enclosing the set $C\mathcal{B}_\varepsilon$, i.e.,

$$\begin{aligned} \text{rad}(C\mathcal{B}_\varepsilon) &= \text{rad}\left(\{Cz \mid z^\top z \leq \varepsilon\}\right) \\ &= \max_{\|z\|_2 \leq \varepsilon} \big\|Cz\big\|_2 = \varepsilon \max_{\|z\|_2 \leq 1} \big\|Cz\big\|_2 = \varepsilon \big\|C\big\|_2, \end{aligned}$$

where $\big\|C\big\|_2$ is the largest singular value of C. □

V. Inner-Approximations

As shown in Sec. IV, the outer-approximation $\widehat{\mathcal{R}}(t)$ computed by Alg. 2 has a Hausdorff distance of at most $\varepsilon_{\max}$ to the exact reachable set $\mathcal{R}(t)$. Consequently, an inner-approximation $\check{\mathcal{R}}(t) \subseteq \mathcal{R}(t)$ can be computed by the Minkowski difference $\check{\mathcal{R}}(t) = \widehat{\mathcal{R}}(t) \ominus \mathcal{B}_\varepsilon$ of the outer-approximation and the hyperball $\mathcal{B}_\varepsilon$ with radius $\varepsilon = \varepsilon_{\max}$. Note that one can also replace $\varepsilon_{\max}$ by the computed error from Alg. 2 to obtain a tighter inner-approximation. Unfortunately, there exists no closed formula for the Minkowski difference of a zonotope and a hyperball. Therefore, we first enclose $\mathcal{B}_\varepsilon$ with a polytope $\mathcal{P} \supseteq \mathcal{B}_\varepsilon$ since the Minkowski difference of a zonotope and a polytope can be computed efficiently if the resulting set is represented by a constrained zonotope:

Proposition 6 (Minkowski difference): *Given a zonotope* $\mathcal{Z} = \langle c, G \rangle_Z \subset \mathbb{R}^n$ *and a polytope* $\mathcal{P} = \langle [v_1 \ \dots \ v_s] \rangle_V \subset \mathbb{R}^n$, *their Minkowski difference can be represented by the constrained zonotope*

$$\mathcal{Z} \ominus \mathcal{P} = \langle c - v_1, [G \ \mathbf{0}], A, b \rangle_{CZ},$$

where

$$A = \begin{bmatrix} G & -G & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ G & \mathbf{0} & \dots & -G \end{bmatrix}, \quad b = \begin{bmatrix} v_1 - v_2 \\ \vdots \\ v_1 - v_s \end{bmatrix}.$$

Proof. According to [50, Lemma 1], the Minkowski difference with a polytope as minuend can be computed as

$$\mathcal{Z} \ominus \mathcal{P} = \bigcap_{i \in \{1,...,s\}} (\mathcal{Z} - v_i) = (\mathcal{Z} - v_1) \cap \ldots \cap (\mathcal{Z} - v_s).$$

Using the equation for the intersection of constrained zonotopes in [44, Eq. (13)], we obtain for the first intersection

$$\begin{aligned} &(\mathcal{Z} - v_1) \cap (\mathcal{Z} - v_2) \\ &\quad = \langle c - v_1, G, [\,], [\,] \rangle_{CZ} \cap \langle c - v_2, G, [\,], [\,] \rangle_{CZ} \\ &\quad \overset{\text{[44, Eq. (13)]}}{=} \langle c - v_1, G, [G \ \ -G], v_1 - v_2 \rangle_{CZ}. \end{aligned}$$

Repeated application of [44, Eq. (13)] yields the claim. □

For general polytopes the number of vertices increases exponentially with the system dimension. To keep the computational complexity small, we enclose the hyperball $\mathcal{B}_\varepsilon$ by a cross-polytope $\langle \varepsilon \sqrt{n} \, [-I_n \ I_n] \rangle_V \supseteq \mathcal{B}_\varepsilon$, which is a special type of polytope with only $2n$ vertices. Since the Hausdorff distance between the hyperball and the enclosing cross-polytope is $(\sqrt{n} - 1)\varepsilon$, we scale the error bound $\varepsilon_{\max}$ by the factor $1/\sqrt{n}$ before executing Alg. 2 in order to obtain an inner-approximation with a maximum Hausdorff distance of $\varepsilon_{\max}$ to the exact reachable set.

VI. Automated Verification

One core application of reachability analysis is the verification of safety specifications. Based on our automated parameter tuning approach, we introduce a fully automated verification algorithm for linear systems, which iteratively refines the tightness of the reachable set inner-approximation and outer-approximation until a given specification can be verified or falsified. We consider specifications of the form

$$\forall t \in [0, t_{\text{end}}]: \left(\bigwedge_{i=1}^{r} \mathcal{R}(t) \subseteq \mathcal{G}_i \right) \wedge \left(\bigwedge_{i=1}^{w} \mathcal{R}(t) \cap \mathcal{F}_i = \emptyset \right)$$

defined by a list of safe sets $\{\mathcal{G}_1, \dots, \mathcal{G}_r\} \subset \mathbb{R}^n$ and a list of unsafe sets $\{\mathcal{F}_1, \dots, \mathcal{F}_w\} \subset \mathbb{R}^n$, both specified as polytopes in halfspace representation. While we omit the dependence on time here for simplicity, the extension to time-varying safe sets and unsafe sets is straightforward. To check if the reachable set satisfies the specification, we need to perform containment and intersection checks on zonotopes and constrained zonotopes:

Proposition 7 (Containment check): *Given a polytope $\mathcal{P} = \langle C, d \rangle_H \subset \mathbb{R}^n$ and a constrained zonotope $\mathcal{CZ} = \langle c, G, A, b \rangle_{CZ} \subset \mathbb{R}^n$, we have*

$$\mathcal{CZ} \subseteq \mathcal{P} \Leftrightarrow \underbrace{\max\left(\nu_1, \dots, \nu_a\right)}_{\nu} \leq 0, \tag{45}$$

where each linear program

$$\forall i \in \{1, \dots, a\}: \quad \nu_i = \max_{\alpha \in \mathbb{R}^\gamma} \; C_{(i,\cdot)} c + C_{(i,\cdot)} G \alpha - d_{(i)} \quad \text{s.t.} \;\; \alpha \in [-\mathbf{1}, \mathbf{1}], \; A\alpha = b.$$

computes the distance to a single polytope halfspace.

Proof. In general, a set $\mathcal{S} \subset \mathbb{R}^n$ is contained in a polytope if it is contained in all polytope halfspaces. The linear program above evaluates the support function (see [14, Def. 1]) of $\mathcal{S}$ along the normal vector of each halfspace, which has to be smaller or equal to the corresponding offset to prove containment [51, Corollary 13.1.1]. □

For a zonotopic in-body $\mathcal{Z} = \langle c, G \rangle_Z \subset \mathbb{R}^n$, there is a closed-form solution ([51, Corollary 13.1.1] with [14, Prop. 1]):

$$\mathcal{Z} \subseteq \mathcal{P} \Leftrightarrow \underbrace{\max\left(Cc - d + \sum_{i=1}^{\gamma} |CG_{(\cdot,i)}| \right)}_{\nu} \leq 0. \tag{46}$$

Next, we consider intersection checks:

Proposition 8 (Intersection check): *A polytope $\mathcal{P} = \langle C, d \rangle_H \subset \mathbb{R}^n$ and a constrained zonotope $\mathcal{CZ} = \langle c, G, A, b \rangle_{CZ} \subset \mathbb{R}^n$ intersect if $\nu \leq 0$ computed by the linear program*

$$\nu = \min_{x \in \mathbb{R}^n, \, \alpha \in \mathbb{R}^\gamma, \, \delta \in \mathbb{R}} \delta \quad \text{s.t.} \;\; \forall i \in \{1, ..., a\}: C_{(i,\cdot)} x - d_{(i)} \leq \delta, \;\; x = c + G\alpha, \;\; A\alpha = b, \;\; \alpha \in [-\mathbf{1}, \mathbf{1}].$$

Proof. If $\forall i \in \{1, ..., a\}: C_{(i,\cdot)} x - d_{(i)} \leq \delta \leq 0$, then there exists a point $x \in \mathcal{CZ}$ that is also contained in $\mathcal{P}$. □

Algorithm 3 Automated verification

Require: Linear system $\dot{x} = Ax + Bu + p$, initial set $\mathcal{X}^0 = \langle c_x, G_x \rangle_Z$, input set $\mathcal{U} = \langle c_u, G_u \rangle_Z$, time horizon t_{end}, specification defined by a list of safe sets $\mathcal{G}_1, \dots, \mathcal{G}_r$ and a list of unsafe sets $\mathcal{F}_1, \dots, \mathcal{F}_w$
Ensure: Specification satisfied (true) or violated (false)

1: $\varepsilon_{\max} \leftarrow$ estimated from simulations
2: **repeat**
3: $\quad \widehat{\mathcal{R}}(t) \leftarrow$ comp. with Alg. 2 using error bound $\varepsilon_{\max}$
4: $\quad \check{\mathcal{R}}(t) \leftarrow \widehat{\mathcal{R}}(t) \ominus \mathcal{B}_\varepsilon$ ▷ see Sec. V
5: $\quad \widehat{\nu}_G, \check{\nu}_G \leftarrow -\infty$, $\widehat{\nu}_F, \check{\nu}_F \leftarrow \infty$
6: $\quad$ **for** $j \leftarrow 1$ to r **do**
7: $\quad\quad \nu \leftarrow$ distance from $\widehat{\mathcal{R}}(t) \subseteq \mathcal{G}_j$ ▷ see (46)
8: $\quad\quad \widehat{\nu}_G \leftarrow \max(\widehat{\nu}_G, \nu)$
9: $\quad\quad \nu \leftarrow$ distance from $\check{\mathcal{R}}(t) \subseteq \mathcal{G}_j$ ▷ see (45)
10: $\quad\quad \check{\nu}_G \leftarrow \max(\check{\nu}_G, \nu)$
11: $\quad$ **end for**
12: $\quad$ **for** $j \leftarrow 1$ to w **do**
13: $\quad\quad \nu \leftarrow$ distance from $\widehat{\mathcal{R}}(t) \cap \mathcal{F}_j = \emptyset$ ▷ see Prop. 8
14: $\quad\quad \widehat{\nu}_F \leftarrow \min(\widehat{\nu}_F, \nu)$
15: $\quad\quad \nu \leftarrow$ distance from $\check{\mathcal{R}}(t) \cap \mathcal{F}_j = \emptyset$ ▷ see Prop. 8
16: $\quad\quad \check{\nu}_F \leftarrow \min(\check{\nu}_F, \nu)$
17: $\quad$ **end for**
18: $\quad \nu \leftarrow \min(-\check{\nu}_G, \check{\nu}_F)$
19: $\quad$ **if** $\widehat{\nu}_G \geq 0$ **then**
20: $\quad\quad \nu \leftarrow \min(\nu, \widehat{\nu}_G)$
21: $\quad$ **end if**
22: $\quad$ **if** $\widehat{\nu}_F \leq 0$ **then**
23: $\quad\quad \nu \leftarrow \min(\nu, -\widehat{\nu}_F)$
24: $\quad$ **end if**
25: $\quad \varepsilon_{\max} \leftarrow \max\left(0.1\,\varepsilon_{\max}, \min(\nu, 0.9\,\varepsilon_{\max})\right)$
26: **until** $\left((\widehat{\nu}_G \leq 0) \wedge (\widehat{\nu}_F > 0)\right) \vee (\check{\nu}_G > 0) \vee (\check{\nu}_F \leq 0)$
27: **return** $(\widehat{\nu}_G \leq 0) \wedge (\widehat{\nu}_F > 0)$

Since a zonotope is just a special case of a constrained zonotope, Prop. 8 can also be used to check if a zonotope intersects a polytope. For both Prop. 7 and Prop. 8, ν is a good estimate for the Hausdorff distance between the sets if polytopes with normalized halfspace normal vectors are used. We utilize this in our automated verification algorithm to estimate the accuracy that is required to verify or falsify the specification.

The overall verification algorithm is summarized in Alg. 3: We first obtain an initial guess for the error bound $\varepsilon_{\max}$ in Line 1 by simulating trajectories for a finite set of points from $\mathcal{X}^0$. The repeat-until loop (Lines 2-26) then refines the inner- and outer-approximations of the reachable set by iteratively decreasing the error bound $\varepsilon_{\max}$ until the specifications can be verified or falsified. In particular, we first compute the outer- and inner-approximation (Lines 3-4). Next, we perform the containment and intersection checks with the safe and unsafe sets (Lines 6-17) and store the corresponding distances $\widehat{\nu}_G, \check{\nu}_G, \widehat{\nu}_F, \check{\nu}_F$. Using these distances, we determine the minimum distance (Lines 18-24), which is then used to

update the error bound $\varepsilon_{\max}$ (Line 25). Since ν is only an estimate, we also restrict the updated error bound to the interval $[0.1\,\varepsilon_{\max}, 0.9\,\varepsilon_{\max}]$ to guarantee convergence and avoid values that are too small. Finally, the specifications are satisfied if the outer-approximation of the reachable set is contained in all safe sets ($\widehat{\nu}_G \leq 0$) and does not intersect any unsafe sets ($\widehat{\nu}_F > 0$). On the other hand, the specifications are falsified if the inner-approximation of the reachable set is not contained in all safe sets ($\check{\nu}_G > 0$) or intersects an unsafe set ($\check{\nu}_F \leq 0$). Improvements for Alg. 3 which we omitted here for simplicity include using the computed error from Alg. 2 instead of the error bound $\varepsilon_{\max}$, omitting re-computation as well as containment and intersection checks for time intervals that are already verified, and only computing inner-approximations if the corresponding outer-approximation is not yet verified.

The space complexity of Alg. 3 is dominated by the inner-approximation $\check{\mathcal{R}}(\tau_k)$, which is $\mathcal{O}(n^4)$ following Prop. 6. Assuming a conservative bound of $\mathcal{O}(p^{3.5})$ for a linear program with p variables according to [52], the runtime complexity of Alg. 3 is $\mathcal{O}(n^7)$ since we evaluate linear programs in Props. 7-8 with $\mathcal{O}(n^2)$ variables, respectively.

VII. Numerical Examples

Let us now demonstrate the performance of our adaptive tuning approach and our verification algorithm. We integrated both algorithms into the MATLAB toolbox CORA [34], and they will be made publicly available with the 2023 release[1]. All computations are carried out on a 2.59GHz quad-core i7 processor with 32GB memory.

A. Electrical Circuit

To showcase the general concept of our approach, we first consider the deliberately simple example of an electric circuit consisting of a resistance $R = 2\Omega$, a capacitor with capacity $C = 1.5\text{mF}$, and a coil with inductance $L = 2.5\text{mH}$:

$$\begin{bmatrix}\dot{u}_C(t)\\ \dot{i}_L(t)\end{bmatrix} = \begin{bmatrix}-\frac{1}{RC} & \frac{1}{C}\\ -\frac{1}{L} & 0\end{bmatrix}\begin{bmatrix}u_C(t)\\ i_L(t)\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{L}\end{bmatrix} u_I(t),$$

where the state is defined by the voltage at the capacitor $u_C(t)$ and the current at the coil $i_L(t)$. The initial set is $\mathcal{X}^0 = [1,3]\text{V} \times [3,5]\text{A}$, the input voltage to the circuit $u_I(t)$ is uncertain within the set $\mathcal{U} = [-0.1, 0.1]\text{V}$, and the time horizon is $t_{\text{end}} = 2\text{s}$. As shown in Fig. 5, the inner- and outer-approximations computed using Alg. 2 and Sec. V converge to the exact reachable set with decreasing error bounds. The computation times are 0.26s for $\varepsilon_{\max} = 0.04$, 0.41s for $\varepsilon_{\max} = 0.02$, and 0.55s for $\varepsilon_{\max} = 0.01$.

B. ARCH Benchmarks

Next, we evaluate our verification algorithm on benchmarks from the 2021 ARCH competition [53], where state-of-the-art reachability tools compete with one another to solve challenging verification tasks. We consider all linear continuous-time systems, which are the building benchmark (BLD) describing the movement of an eight-story hospital building, the International Space Station (ISS) benchmark modeling a service module of the ISS, the Heat 3D benchmark (HEAT) representing a spatially discretized version of the heat equation, and the clamped beam benchmark (CB) monitoring oscillations of a beam. The results in Tab. I demonstrate that our fully automated verification algorithm correctly verifies all safe benchmarks without being significantly slower than state-of-the-art tools that require extensive parameter tuning by experts. Please note that we compare only to the computation time of other tools achieved by the optimal run with expert-tuned algorithm parameters, disregarding the significant amount of time required for tuning. Moreover, our algorithm also successfully falsifies the two unsafe benchmarks, where our computation time is slightly worse because the other tools do not explicitly falsify these benchmarks but only test if they cannot be verified, which is considerably easier.

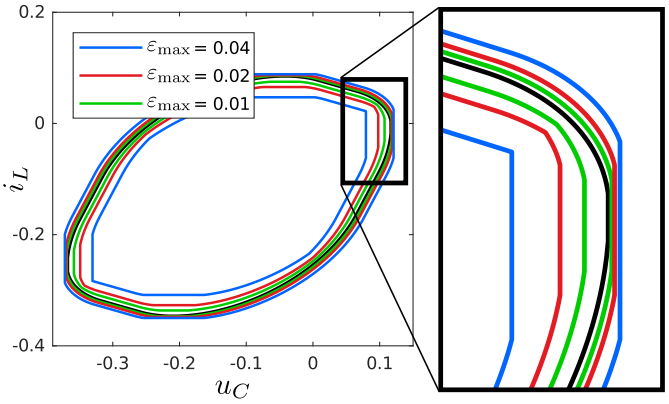


Fig. 5. Inner- and outer-approximations of the final reachable set $\mathcal{R}(t_{\text{end}})$ for the electric circuit using different error bounds $\varepsilon_{\max}$, with the exact reachable set is shown in black.

C. Autonomous Car

Finally, we show that our verification algorithm can handle complex verification tasks featuring time-varying specifications. To this end, we consider the benchmark proposed in [54], where the task is to verify that a planned reference trajectory $x_{\text{ref}}(t)$ tracked by a feedback controller is robustly safe despite disturbances and measurement errors. The nonlinear vehicle model in [54, Eq. (3)] is replaced by a linear point mass model, which yields the closed-loop system

$$\begin{bmatrix}\dot{x}(t)\\ \dot{x}_{\text{ref}}(t)\end{bmatrix} = \begin{bmatrix}A+BK & -BK\\ \mathbf{0} & A\end{bmatrix}\begin{bmatrix}x(t)\\ x_{\text{ref}}(t)\end{bmatrix} + \begin{bmatrix}B & B & BK\\ B & \mathbf{0} & \mathbf{0}\end{bmatrix} u(t)$$

with $A = [\mathbf{0}\ [I_2\ \mathbf{0}]^\top]$, $B = [\mathbf{0}\ I_2]^\top$, and feedback matrix $K \in \mathbb{R}^{2\times 4}$. The initial set is $\mathcal{X}^0 = (x_0 + \mathcal{V}) \times x_0$ and the set of uncertain inputs is $\mathcal{U} = u_{\text{ref}}(t) \times \mathcal{W} \times \mathcal{V}$, where $\mathcal{W} \subset \mathbb{R}^2$ and $\mathcal{V} \subset \mathbb{R}^4$ are the sets of disturbances and measurement errors taken from [54, Sec. 3], and the initial state $x_0 \in \mathbb{R}^4$ and control inputs for the reference trajectory $u_{\text{ref}}(t) \in \mathbb{R}^2$ are specific to the considered traffic scenario. To compute occupied space of the car, we apply affine arithmetic [55] to evaluate the nonlinear map in [54, Eq. (4)], where we determine the orientation of the car from the direction of the velocity vector.

For verification, we consider the traffic scenario *BEL_Putte-*

[1]available at `https://cora.in.tum.de`

TABLE I

COMPARISON OF COMPUTATION TIMES ON THE ARCH BENCHMARKS, WHERE n IS THE SYSTEM DIMENSION, m IS THE NUMBER OF INPUTS, AND ℓ IS THE OUTPUT DIMENSION. FOR OUR APPROACH WE ADDITIONALLY SPECIFY THE NUMBER OF REFINEMENT ITERATIONS OF ALG. 3. THE COMPUTATION TIMES OF THE OTHER TOOLS ARE TAKEN FROM [53].

Benchmark					Our approach		Time comparison			
Identifier	n	m	ℓ	Safe?	Time	Iterations	CORA	HyDRA	JuliaReach	SpaceEx
HEAT01	125	0	1	✓	2.2s	2	2.2s	13.2s	0.13s	4.2s
HEAT02	1000	0	1	✓	59s	1	9.3s	160s	32s	—
CBC01	201	0	1	✓	28s	1	7.1s	—	1.4s	312.78s
CBF01	200	1	1	✓	144s	2	30s	—	12s	318.88s
BLDC01-BDS01	49	0	1	✓	1.7s	1	2.9s	0.426s	0.0096s	1.6s
BLDF01-BDS01	48	1	1	✓	2.1s	1	3.3s	—	0.012s	1.8s
ISSC01-ISS02	273	0	3	✓	4.3s	1	1.3s	—	1.4s	29s
ISSC01-ISU02	273	0	3	✗	10s	4	0.072s	—	1.4s	29s
ISSF01-ISS01	270	3	3	✓	75s	2	59s	—	10s	49s
ISSF01-ISU01	270	3	3	✗	191s	3	38s	—	10s	48s

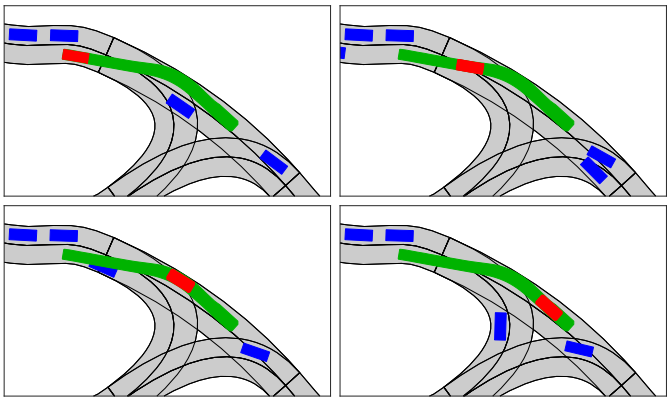


Fig. 6. Traffic scenario at times 0s, 1s, 2s, and 3s, where the reachable set for the whole time horizon, the reachable set for the current time point, and the other traffic participants are shown.

4_2_T-1 from the CommonRoad database[2]. The unsafe sets $\mathcal{F}_i$ for the verification task are given by the road boundary and the occupancy space of other traffic participants. Since the road boundary is non-convex, we use triangulation to represent it as the union of 460 convex polytopes. The occupancy spaces of other traffic participants over time intervals of length 0.1s are represented by polytopes, which results in 170 time-varying unsafe sets for the six vehicles in the scenario. The safe set $\mathcal{G}_i$ is given by the constraint that the absolute acceleration should stay below $11.5\,\mathrm{m\,s^{-2}}$, which we inner-approximate by a polytope with 20 halfspaces. Even for this complex verification task, Alg. 3 only requires 64s and two refinements of the error bound $\varepsilon_{\max}$ to prove that the reference trajectory is robustly safe (see Fig. 6).

VIII. DISCUSSION

Despite the convincing results of our automated verification algorithm in Sec. VII, there is still potential for improvement: According to Tab. I, other reachability tools solve high-dimensional benchmarks often faster than our approach since they apply tailored algorithms, such as block-decomposition [18] and Krylov subspace methods [19], [20]. Therefore, a natural next step is to extend our concept of automated parameter tuning via error analysis to these specialized algorithms to accelerate the verification of high-dimensional systems. In addition, since many reachability algorithms for nonlinear systems [56], [57] are based on reachability analysis for linear systems, another intriguing research direction is the extension to nonlinear systems. Verification algorithms based on implicit set representations, such as support functions [58], also present an interesting comparison to our proposed method, which computes explicit sets.

Moreover, for systems where some states have no initial uncertainty and are not influenced by uncertain inputs, our proposed algorithm cannot falsify the system since the computed inner-approximation of the reachable set will always be empty. An example is the autonomous car in Sec. VII-C, where the states corresponding to the reference trajectory are not subject to any uncertainty. Fortunately, these cases are easy to detect and one can use classical safety falsification techniques, such as Monte Carlo methods [59], Bayesian optimization [60], or cross-entropy techniques [61] instead. In general, combining safety falsification methods with our algorithm may accelerate the falsification process and provide the user with a concrete counterexample in the form of a falsifying trajectory.

Finally, while our algorithm already supports the general case of specifications defined by time-varying safe sets and unsafe sets, future work could include an extension to temporal logic specifications. This may be realized by a conversion to reachset temporal logic [62], a particular type of temporal logic that can be evaluated on reachable sets directly. Another possibility is to convert temporal logic specifications to an acceptance automaton [63], which can then be combined with the linear system via parallel composition [64].

IX. CONCLUSION

In this work, we propose a paradigm shift for reachability analysis of linear systems: Instead of requiring the user to manually tune algorithm parameters such as the time step size, our approach automatically adapts all parameters such that the

[2]available at `https://commonroad.in.tum.de/scenarios`

computed outer-approximation respects a desired maximum distance to the exact reachable set. Building on this result, we then extract an inner-approximation of the reachable set directly from the outer-approximation using the Minkowski difference, which finally enables us to design a sound verification algorithm that automatically refines the inner- and outer-approximations until specifications given by time-varying safe and unsafe sets can either be verified or falsified. An evaluation on benchmarks representing the current limits for state-of-the-art reachability tools demonstrates that our approach is competitive regarding the computation time, even for high-dimensional systems. Overall, the autonomy of our approach enables non-experts to verify or falsify safety specifications for linear systems in reasonable time.

Appendix A: Additional Propositions

For the proof of Prop. 1 we require the error induced by outer-approximating the linear combination for zonotopes:

Proposition 9: *Given the homogeneous solution* $\mathcal{H}(t_k) = \langle c_h, G_h \rangle_Z$ *with* $G_h \in \mathbb{R}^{n \times \gamma_h}$ *and the particular solution* $\mathcal{P}^u(\Delta t_k) = c_p$*, the Hausdorff distance between the exact linear combination*

$$\mathcal{S} = \operatorname{comb}\big(\mathcal{H}(t_k), \mathcal{H}(t_{k+1})\big)$$

with $\mathcal{H}(t_{k+1}) = e^{A\Delta t_k}\mathcal{H}(t_k) + \mathcal{P}^u(\Delta t_k)$ *and the corresponding zonotope outer-approximation* $\widehat{\mathcal{S}}$ *computed using* (8) *is bounded by*

$$d_H\big(\mathcal{S}, \widehat{\mathcal{S}}\big) \leq \sqrt{\gamma_h}\, \big\|G_h^{(-)}\big\|_2,$$

where $G_h^{(-)} = (e^{A\Delta t_k} - I_n)G_h$.

Proof. We insert the homogeneous and particular solutions into (5) and shift the interval of λ from $[0, 1]$ to $[-1, 1]$, which yields

$$\begin{aligned}
\mathcal{S} = \Bigg\{ & \lambda\Big(c_h + \sum_{i=1}^{\gamma_h} G_{h(\cdot,i)}\alpha_i\Big) + (1-\lambda)\Big(e^{A\Delta t_k} \\
& \Big(c_h + \sum_{i=1}^{\gamma_h} G_{h(\cdot,i)}\alpha_i\Big) + c_p\Big) \;\Big|\; \alpha_i \in [-1,1], \lambda \in [0,1]\Bigg\}, \\
= \Bigg\{ & 0.5\Big(\big(I_n + e^{A\Delta t_k}\big)c_h + c_p\Big) \\
& + 0.5\lambda\Big(\big(e^{A\Delta t_k} - I_n\big)c_h + c_p\Big) + 0.5\sum_{i=1}^{\gamma_h} G_{h(\cdot,i)}^{(+)}\alpha_i \\
& + 0.5\sum_{i=1}^{\gamma_h} G_{h(\cdot,i)}^{(-)}\alpha_i\lambda \;\Big|\; \alpha_i, \lambda \in [-1,1]\Bigg\},
\end{aligned}$$

with $G_h^{(+)} = (e^{A\Delta t_k} + I_n)G_h$ and $G_h^{(-)} = (e^{A\Delta t_k} - I_n)G_h$. In order to represent this exact linear combination $\mathcal{S}$ as a zonotope, we have to substitute the bilinear factors $\alpha_i\lambda$ in the last term by additional linear factors $\omega_i \in [-1, 1]$. By neglecting the dependency between α and λ, we obtain the outer-approximation $\widehat{\mathcal{S}}$ of the exact linear combination $\mathcal{S}$. Due to the obvious containment $\mathcal{S} \subseteq \widehat{\mathcal{S}}$, the formula for the Hausdorff distance in (9) simplifies to

$$d_H\big(\mathcal{S}, \widehat{\mathcal{S}}\big) = \max_{\widehat{s}\in\widehat{\mathcal{S}}} \min_{s\in\mathcal{S}} \big\|\widehat{s} - s\big\|_2.$$

Exploiting the identical factors before and after conversion, all terms but one cancel out and we obtain

$$\begin{aligned}
\max_{\widehat{s}\in\widehat{\mathcal{S}}} \min_{s\in\mathcal{S}} \big\|\widehat{s} - s\big\|_2 &\leq \max_{\substack{\omega_i\in[-1,1]\\ \alpha_i\in[-1,1]\\ \lambda\in[-1,1]}} 0.5\Big\|\sum_{i=1}^{\gamma_h} G_{h(\cdot,i)}^{(-)}(\omega_i - \alpha_i\lambda)\Big\|_2 \\
&\leq 0.5\big\|G_h^{(-)}\big\|_2 \max_{\varphi\in[-\mathbf{1},\mathbf{1}]} \big\|(\omega - \alpha\lambda)\big\|_2 \qquad (47)
\end{aligned}$$

with $\alpha = [\alpha_1 \;\ldots\; \alpha_{\gamma_h}]^\top$, $\omega = [\omega_1 \;\ldots\; \omega_{\gamma_h}]^\top$, and $\varphi = [\omega \;\alpha\; \lambda]^\top$. According to the Bauer Maximum Principle, we may assume that the maximum is attained at a point which satisfies the constraints

$$\forall i \in \{1, ..., \gamma_h\} : \omega_i^2 = 1, \alpha_i^2 = 1, \quad \text{and} \quad \lambda^2 = 1$$

for the maximization term in (47). Consequently, we obtain

$$\begin{aligned}
&\max_{\varphi\in[-\mathbf{1},\mathbf{1}]} \big\|\omega - \alpha\lambda\big\|_2^2 = \max_{\varphi\in[-\mathbf{1},\mathbf{1}]} \omega^\top\omega + \alpha^\top\alpha\,\lambda^2 - 2\lambda\,\omega^\top\alpha \\
&\leq \max_{\varphi\in[-\mathbf{1},\mathbf{1}]} \omega^\top\omega + \max_{\varphi\in[-\mathbf{1},\mathbf{1}]} \alpha^\top\alpha\,\lambda^2 + \max_{\varphi\in[-\mathbf{1},\mathbf{1}]} -2\lambda\,\omega^\top\alpha \\
&= \gamma_h + \gamma_h + 2\gamma_h = 4\gamma_h,
\end{aligned}$$

which implies $\max_{\varphi\in[-\mathbf{1},\mathbf{1}]} \big\|\omega - \alpha\lambda\big\|_2 \leq 2\sqrt{\gamma_h}$. We insert this result into (47) to obtain the error

$$d_H\big(\mathcal{S}, \widehat{\mathcal{S}}\big) \leq 0.5\big\|G_h^{(-)}\big\|_2\, 2\sqrt{\gamma_h} = \sqrt{\gamma_h}\big\|G_h^{(-)}\big\|_2,$$

which concludes the proof. □

To derive the error $\Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k, \eta_k)$ for one time step contained in the particular solution $e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)$ as used in Prop. 2, we require the following proposition:

Proposition 10: *For a compact set* $\mathcal{S} \subset \mathbb{R}^n$ *and two matrices* $M_1, M_2 \in \mathbb{R}^{m\times n}$*, we have*

$$d_H\big((M_1 + M_2)\mathcal{S}, M_1\mathcal{S}\big) \leq d_H\big(\mathbf{0}, M_2\mathcal{S}\big).$$

Proof. For the left-hand side, we choose $s_1 = s_2$ for both min-operations in the definition (9) of the Hausdorff distance:

$$\begin{aligned}
& d_H\big((M_1 + M_2)\mathcal{S}, M_1\mathcal{S}\big) \\
&\overset{(9)}{=} \max\Big\{ \max_{s_1\in\mathcal{S}} \Big(\min_{s_2\in\mathcal{S}} \big\|(M_1 + M_2)s_1 - M_1s_2\big\|_2 \Big), \\
&\qquad \max_{s_2\in\mathcal{S}} \Big(\min_{s_1\in\mathcal{S}} \big\|(M_1 + M_2)s_1 - M_1s_2\big\|_2 \Big)\Big\} \\
&\leq \max\Big\{ \max_{s_1\in\mathcal{S}} \big\|(M_1 + M_2)s_1 - M_1s_1\big\|_2, \\
&\qquad \max_{s_2\in\mathcal{S}} \big\|(M_1 + M_2)s_2 - M_1s_2\big\|_2\Big\} \\
&= \max_{s\in\mathcal{S}} \big\|M_2s\big\|_2.
\end{aligned}$$

The right-hand side evaluates trivially to

$$d_H\big(\mathbf{0}, M_2\mathcal{S}\big) = \max_{s\in\mathcal{S}} \big\|M_2s\big\|_2,$$

which combined with the result above yields the claim. □

Using Prop. 10, we obtain the following error bound:

Proposition 11: *The Hausdorff distance between the propa-*

gated exact particular solution

$$e^{At_k}\mathcal{P}^{\mathcal{U}}(\Delta t_k) = \left\{ e^{At_k} \int_0^{\Delta t_k} e^{A(\Delta t_k-\theta)} u(\theta)\,\mathrm{d}\theta \;\middle|\; u(\theta) \in \mathcal{U}_0 \right\}$$

and the outer-approximation $e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)$ *with* $\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)$ *from* (20) *is bounded by* (25).

Proof. We obtain a tight bound for the error by computing the distance between an inner-approximation $\check{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)$ and the outer-approximation $\widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)$ in (20). By considering uncertain but constant inputs we compute an inner-approximation $\check{\mathcal{P}}^{\mathcal{U}}(\Delta t_k) \subseteq \mathcal{P}^{\mathcal{U}}(\Delta t_k)$ as

$$\begin{aligned}\check{\mathcal{P}}^{\mathcal{U}}(\Delta t_k) &= \int_0^{\Delta t_k} e^{A(\Delta t_k-\theta)}\mathrm{d}\theta\,\mathcal{U}_0 \\ &= \left(\sum_{i=0}^{\infty} \frac{A^i \Delta t_k^{i+1}}{(i+1)!}\right)\mathcal{U}_0 = \left(\Delta t_k\, I_n + \sum_{i=1}^{\infty} \tilde{A}_i\right)\mathcal{U}_0\end{aligned}$$

where $\tilde{A}_i = \frac{A^i \Delta t_k^{i+1}}{(i+1)!}$. Applying Prop. 10 with $M_1 = \Delta t_k I_n$ and $M_2 = \sum_{i=1}^{\infty} \tilde{A}_i$, we have

$$\begin{aligned} d_H\big(\check{\mathcal{P}}^{\mathcal{U}}(\Delta t_k), \Delta t_k\,\mathcal{U}_0\big) &\le d_H\left(\mathbf{0}, \left(\sum_{i=1}^{\infty}\tilde{A}_i\right)\mathcal{U}_0\right) \\ &\overset{(12)}{\le} \operatorname{err}\left(\left(\sum_{i=1}^{\infty}\tilde{A}_i\right)\mathcal{U}_0\right) \\ &\overset{(18)}{\le} \operatorname{err}\left(\left(\sum_{i=1}^{\eta}\tilde{A}_i\right)\mathcal{U}_0 \oplus \boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k)\Delta t_k\,\mathcal{U}_0\right).\end{aligned}$$

From (20), we have the trivial containment $\Delta t_k\,\mathcal{U}_0 \subset \widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)$ and thus

$$\begin{aligned} &d_H\big(\Delta t_k\,\mathcal{U}, \widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)\big) \\ &\quad\le \operatorname{err}\left(\bigoplus_{i=1}^{\eta}\left(\tilde{A}_i\,\mathcal{U}_0\right) \oplus \boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k)\Delta t_k\,\mathcal{U}_0\right).\end{aligned}$$

We use the triangle inequality to combine the last two results:

$$\begin{aligned} d_H\big(\mathcal{P}^{\mathcal{U}}(\Delta t_k), \widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)\big) &\le d_H\big(\check{\mathcal{P}}^{\mathcal{U}}(\Delta t_k), \widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)\big) \\ &\le d_H\big(\check{\mathcal{P}}^{\mathcal{U}}(\Delta t_k), \Delta t_k\,\mathcal{U}_0\big) + d_H\big(\Delta t_k\,\mathcal{U}_0, \widehat{\mathcal{P}}^{\mathcal{U}}(\Delta t_k)\big) \\ &\le \operatorname{err}\left(\left(\sum_{i=1}^{\eta}\tilde{A}_i\right)\mathcal{U}_0 \oplus \boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k)\Delta t_k\,\mathcal{U}_0\right) \\ &\quad + \operatorname{err}\left(\bigoplus_{i=1}^{\eta}\left(\tilde{A}_i\,\mathcal{U}_0\right) \oplus \boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k)\Delta t_k\,\mathcal{U}_0\right).\end{aligned}$$

Including the mapping by e^{At_k} then yields the final result. □

Appendix B: Additional Lemmata

For the proof of Theorem 1, we require results about the limit behavior of the error terms, which we derive here. First, we examine the remainder of the exponential matrix:

Lemma 1: *The remainder of the exponential matrix* $\boldsymbol{\mathcal{E}}(\Delta t_k, \eta_k)$ *in* (18) *satisfies*

$$\lim_{\Delta t_k \to 0} \boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k) = \mathbf{0} \quad \textit{and} \quad \boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k) \sim \mathcal{O}\left(\Delta t_k^{\eta_k+1}\right).$$

Proof. For the limit value we obtain from (18)

$$\begin{aligned}\lim_{\Delta t_k\to 0} E(\Delta t_k,\eta_k) &= e^{|A|0} - \sum_{i=0}^{\eta_k}\frac{1}{i!}\big(|A|0\big)^i = \mathbf{0}, \\ \lim_{\Delta t_k\to 0} \boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k) &= \lim_{\Delta t_k\to 0}[-E(\Delta t_k,\eta_k), E(\Delta t_k,\eta_k)] = \mathbf{0},\end{aligned}$$

and the asymptotic behavior follows from

$$E(\Delta t_k,\eta_k) = e^{|A|\Delta t_k} - \sum_{i=0}^{\eta_k}\frac{\big(|A|\Delta t_k\big)^i}{i!} = \sum_{i=\eta_k+1}^{\infty}\frac{\big(|A|\Delta t_k\big)^i}{i!},$$

which yields $\boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k) \sim \mathcal{O}\left(\Delta t_k^{\eta_k+1}\right)$. □

Next, we consider the error of the particular solution:

Lemma 2: *The error* $\Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k,\eta_k)$ *in* (25) *satisfies*

$$\lim_{\Delta t_k\to 0}\Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k,\eta_k) = 0 \quad \textit{and} \quad \Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k,\eta_k) \sim \mathcal{O}\big(\Delta t_k^2\big).$$

Proof. Using the definition of the error $\Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k,\eta_k)$ according to (25) with $\tilde{A}_i = \frac{A^i\Delta t_k^{i+1}}{(i+1)!}$ and Lemma 1, we have

$$\begin{aligned}&\lim_{\Delta t_k\to 0}\tilde{A}_i = 0, && \tilde{A}_i \sim \mathcal{O}\big(\Delta t_k^2\big), \\ &\lim_{\Delta t_k\to 0}\boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k)\Delta t_k = 0, && \boldsymbol{\mathcal{E}}(\Delta t_k,\eta_k)\Delta t_k \sim \mathcal{O}\left(\Delta t_k^{\eta_k+2}\right),\end{aligned}$$

where we exploit that the minimal index is $i = 1$. It is then straightforward to obtain the limit and asymptotic behavior for $\Delta\varepsilon_k^{\mathcal{U}}(\Delta t_k,\eta_k)$. □

While we require a faster than linear decrease in Δt_k for the accumulating error, a linear decrease is sufficient for the non-accumulating error as it only affects a single time step:

Lemma 3: *The error* $\Delta\varepsilon_k^{h}(\Delta t_k,\eta_k)$ *in* (23) *satisfies*

$$\lim_{\Delta t_k\to 0}\Delta\varepsilon_k^{h}(\Delta t_k,\eta_k) = 0 \quad \textit{and} \quad \Delta\varepsilon_k^{h}(\Delta t_k,\eta_k) \sim \mathcal{O}(\Delta t_k).$$

Proof. We examine the two individual terms of $\Delta\varepsilon_k^{\mathrm{h}}(\Delta t_k,\eta_k) = 2\operatorname{err}(\mathcal{C}) + \sqrt{\gamma_h}\left\|G_h^{(-)}\right\|_2$ separately:

$$\begin{aligned}\lim_{\Delta t_k\to 0}\sqrt{\gamma_h}\left\|G_h^{(-)}\right\|_2 &= \lim_{\Delta t_k\to 0}\sqrt{\gamma_h}\left\|(e^{A\Delta t_k} - I_n)G_h\right\|_2 \\ &= \sqrt{\gamma_h}\left\|(e^{\mathbf{0}} - I_n)G_h\right\|_2 = \sqrt{\gamma_h}\left\|\mathbf{0}\right\|_2 = 0.\end{aligned}$$

To analyze the asymptotic behavior, it suffices to look at $(e^{A\Delta t_k} - I_n)G_h$ as the matrix G_h and the factor γ_h do not depend on the time step size: Since $(e^{A\Delta t_k} - I_n) \sim \mathcal{O}(\Delta t_k)$, we consequently obtain

$$\sqrt{\gamma_h}\left\|G_h^{(-)}\right\|_2 \sim \mathcal{O}(\Delta t_k).$$

For the limit behavior of the term $2\operatorname{err}(\mathcal{C})$, we have

$$\begin{aligned}\lim_{\Delta t_k\to 0}\mathcal{I}_i(\Delta t_k) &\overset{(17)}{=} \mathcal{I}_i(0) = \left[\big(i^{\frac{-i}{i-1}} - i^{\frac{-1}{i-1}}\big)0^i, 0\right] = 0, \\ \lim_{\Delta t_k\to 0}\boldsymbol{\mathcal{F}}(\Delta t_k,\eta_k) &\overset{(15)}{=} \bigoplus_{i=2}^{\eta_k}\mathcal{I}_i(0)\,\frac{A^i}{i!} \oplus \boldsymbol{\mathcal{E}}(0,\eta_k) = \mathbf{0}, \\ \lim_{\Delta t_k\to 0}\boldsymbol{\mathcal{G}}(\Delta t_k,\eta_k) &\overset{(16)}{=} \bigoplus_{i=2}^{\eta_k+1}\mathcal{I}_i(0)\,\frac{A^i}{i!} \oplus \boldsymbol{\mathcal{E}}(0,\eta_k)0 = \mathbf{0},\end{aligned}$$

which entails

$$\lim_{\Delta t_k \to 0} 2 \underbrace{(\mathrm{err}(\boldsymbol{\mathcal{F}}(0,\eta)\mathcal{H}(t_k)) + \mathrm{err}(\boldsymbol{\mathcal{G}}(0,\eta)\tilde{u}))}_{\mathrm{err}(\mathcal{C})} = 0.$$

Moreover, we have $\mathcal{I}_i(\Delta t_k) \sim \mathcal{O}\big(\Delta t_k^2\big)$ as the minimal index is $i = 2$, so that we obtain in combination with Lemma 1

$$2(\mathrm{err}(\boldsymbol{\mathcal{F}}(0,\eta)\mathcal{H}(t_k)) + \mathrm{err}(\boldsymbol{\mathcal{G}}(0,\eta)\tilde{u})) \sim \mathcal{O}\big(\Delta t_k^2\big)\,.$$

It is then straightforward to obtain the limit and asymptotic behavior for $\Delta\varepsilon_k^{\mathrm{h}}(\Delta t_k, \eta_k)$. □

Lemma 4: *The error* $\Delta\varepsilon_k^{\mathcal{U},\tau}(\Delta t_k, \eta_k)$ *in* (26) *satisfies*

$$\lim_{\Delta t_k \to 0} \Delta\varepsilon_k^{\mathcal{U},\tau}(\Delta t_k, \eta_k) = 0 \text{ and } \Delta\varepsilon_k^{\mathcal{U},\tau}(\Delta t_k, \eta_k) \sim \mathcal{O}(\Delta t_k)\,.$$

Proof. For the limit, we insert $\Delta t_k = 0$ into (20) to obtain

$$\begin{aligned}\lim_{\Delta t_k \to 0} \Delta\varepsilon_k^{\mathcal{U},\tau}(\Delta t_k, \eta_k) &= \mathrm{err}\Big(e^{At_k}\widehat{\mathcal{P}}^{\mathcal{U}}(0)\Big) \\ &= \mathrm{err}\left(e^{At_k}\left(\bigoplus_{i=0}^{\eta_k} \frac{A^i 0^{i+1}}{(i+1)!}\mathcal{U}_0 \oplus \boldsymbol{\mathcal{E}}(0,\eta_k)0\mathcal{U}_0\right)\right) = 0.\end{aligned}$$

According to Lemma 1, we have $\boldsymbol{\mathcal{E}}(\Delta t_k, \eta_k) \sim \mathcal{O}(\Delta t_k)$, and with the first term of the sum above, we obtain $\Delta\varepsilon_k^{\mathcal{U},\tau}(\Delta t_k, \eta_k) \sim \mathcal{O}(\Delta t_k)$. □

References

[1] G. Frehse, B. H. Krogh, and R. A. Rutenbar, "Verifying analog oscillator circuits using forward/backward abstraction refinement," in *Proc. of the Design Automation & Test in Europe Conference*. IEEE, 2006, pp. 257–262.

[2] H. N. V. Pico and D. C. Aliprantis, "Voltage ride-through capability verification of wind turbines with fully-rated converters using reachability analysis," *IEEE Transactions on Energy Conversion*, vol. 29, no. 2, pp. 392–405, 2014.

[3] S. Lengagne, N. Ramdani, and P. Fraisse, "Planning and fast replanning safe motions for humanoid robots," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1095–1106, 2011.

[4] S. Kaynama et al., "Computing the viability kernel using maximal reachable sets," in *Proc. of the 15th International Conference on Hybrid Systems: Computation and Control*. ACM, 2012, pp. 55–64.

[5] K. Hobbs et al., "Space debris collision detection using reachability," in *Proc. of the 5th International Workshop on Applied Verification of Continuous and Hybrid Systems*, 2018, pp. 218–228.

[6] S. Vaskov et al., "Guaranteed safe reachability-based trajectory design for a high-fidelity model of an autonomous passenger vehicle," in *Proc. of the American Control Conference*, 2019, pp. 705–710.

[7] 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.

[8] A. Vinod, B. HomChaudhuri, and M. Oishi, "Forward stochastic reachability analysis for uncontrolled linear systems using Fourier transforms," in *Proc. of the 20th International Conference on Hybrid Systems: Computation and Control*. ACM, 2017, pp. 35–44.

[9] A. Devonport, F. Yang, L. El Ghaoui, and M. Arcak, "Data-driven reachability analysis with christoffel functions," in *Proc. of the 60th Conference on Decision and Control*, 2021, pp. 5067–5072.

[10] A. Thorpe, K. Ortiz, and M. Oishi, "Learning approximate forward reachable sets using separating kernels," in *Learning for Dynamics and Control*, 2021, pp. 201–212.

[11] A. Girard, "Reachability of uncertain linear systems using zonotopes," in *8th International Workshop on Hybrid Systems: Computation and Control*. Springer, 2005, pp. 291–305.

[12] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Dissertation, Technische Universität München, 2010.

[13] G. Frehse et al., "SpaceEx: Scalable verification of hybrid systems," in *Proc. of the 23rd International Conference on Computer Aided Verification*. Springer, 2011, pp. 379–395.

[14] C. Le Guernic and A. Girard, "Reachability analysis of linear systems using support functions," *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 250–262, 2010.

[15] A. Donzé and O. Maler, "Systematic simulation using sensitivity analysis," in *10th International Workshop on Hybrid Systems: Computation and Control*. Springer, 2007, pp. 174–189.

[16] T. Dang et al., "Sensitive state-space exploration," in *Proc. of the 47th Conference on Decision and Control*. IEEE, 2008, pp. 4049–4054.

[17] S. Kaynama and M. Oishi, "Complexity reduction through a schur-based decomposition for reachability analysis of linear time-invariant systems," *International Journal of Control*, vol. 84, no. 1, pp. 165–179, 2011.

[18] S. Bogomolov et al., "Reach set approximation through decomposition with low-dimensional sets and high-dimensional matrices," in *Proc. of the 21st International Conference on Hybrid Systems: Computation and Control*. ACM, 2018, pp. 41–50.

[19] M. Althoff, "Reachability analysis of large linear systems with uncertain inputs in the Krylov subspace," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 477–492, 2020.

[20] S. Bak, H.-D. Tran, and T. T. Johnson, "Numerical verification of affine systems with up to a billion dimensions," in *Proc. of the 22nd International Conference on Hybrid Systems: Computation and Control*. ACM, 2019, pp. 23–32.

[21] A. Girard, C. Le Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs," in *9th International Workshop on Hybrid Systems: Computation and Control*. Springer, 2006, pp. 257–271.

[22] A. Hamadeh and J. Goncalves, "Reachability analysis of continuous-time piecewise affine systems," *Automatica*, vol. 44, no. 12, pp. 3189–3194, 2008.

[23] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal techniques for reachability analysis," in *3rd International Workshop on Hybrid Systems: Computation and Control*. Springer, 2000, pp. 202–214.

[24] N. Kochdumper and M. Althoff, "Computing non-convex inner-approximations of reachable sets for nonlinear continuous systems," in *Proc. of the 59th Conference on Decision and Control*. IEEE, 2020, pp. 2130–2137.

[25] E. Goubault and S. Putot, "Inner and outer reachability for the verification of control systems," in *Proc. of the 22nd International Conference on Hybrid Systems: Computation and Control*. ACM, 2019, pp. 11–22.

[26] E. Asarin et al., "Approximate reachability analysis of piecewise-linear dynamical systems," in *3rd International Workshop on Hybrid Systems: Computation and Control*. Springer, 2000, pp. 20–31.

[27] M. Althoff and G. Frehse, "Combining zonotopes and support functions for efficient reachability analysis of linear systems," in *Proc. of the 55th Conference on Decision and Control*. IEEE, 2016, pp. 7439–7446.

[28] P. S. Duggirala and M. Viswanathan, "Parsimonious, simulation based verification of linear systems," in *Proc. of the 28th International Conference on Computer Aided Verification*. Springer, 2016, pp. 477–494.

[29] O. Stursberg, A. Fehnker, Z. Han, and B. H. Krogh, "Verification of a cruise control system using counterexample-guided search," *Control Engineering Practice*, vol. 12, no. 10, pp. 1269–1278, 2004.

[30] S. Bogomolov et al., "Guided search for hybrid systems based on coarse-grained space abstractions," *International Journal on Software Tools for Technology Transfer*, vol. 18, pp. 449–467, 2016.

[31] ——, "Counterexample-guided refinement of template polyhedra," in *23rd International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2017, pp. 589–606.

[32] S. Schupp and E. Ábrahám, "Efficient dynamic error reduction for hybrid systems reachability analysis," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2018, pp. 287–302.

[33] T. T. Johnson, S. Bak, M. Caccamo, and L. Sha, "Real-time reachability for verified simplex design," *ACM Transactions on Embedded Computing Systems*, vol. 15, no. 2, 2016.

[34] M. Althoff, "An introduction to CORA 2015," in *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015, pp. 120–151.

[35] X. Chen et al., "Flow*: An analyzer for non-linear hybrid systems," in *Proc. of the 25th International Conference Computer-Aided Verification*. Springer, 2013, pp. 258–263.

[36] S. Schupp et al., "HyPRO: A C++ library of state set representations for hybrid systems reachability analysis," in *NASA Formal Methods Symposium*. Springer, 2017, pp. 288–294.

[37] S. Bak and P. S. Duggirala, "HyLAA: A tool for computing simulation-equivalent reachability for linear systems," in *Proc. of the 20th International Conference on Hybrid Systems: Computation and Control*. ACM, 2017, pp. 173–178.

[38] S. Bogomolov et al., "JuliaReach: a toolbox for set-based reachability," in *Proc. of the 22nd International Conference on Hybrid Systems: Computation and Control*. ACM, 2019, pp. 39–44.

[39] R. Ray et al., "XSpeed: Accelerating reachability analysis on multi-core processors," in *Haifa Verification Conference*. Springer, 2015, pp. 3–18.

[40] S. Bak, S. Bogomolov, and C. Schilling, "High-level hybrid systems analysis with Hypy," in *Proc. of the Workshop on Applied Verification of Continuous and Hybrid Systems*, 2016, pp. 80–90.

[41] P. Prabhakar and M. Viswanathan, "A dynamic algorithm for approximate flow computations," in *Proc. of the 14th International Conference on Hybrid Systems: Computation and Control*. ACM, 2011, pp. 133–142.

[42] G. Frehse, R. Kateja, and C. Le Guernic, "Flowpipe approximation and clustering in space-time," in *Proc. of the 16th International Conference on Hybrid Systems: Computation and Control*. ACM, 2013, pp. 203–212.

[43] M. Wetzlinger, N. Kochdumper, and M. Althoff, "Adaptive parameter tuning for reachability analysis of linear systems," in *Proc. of the 59th Conference on Decision and Control*. IEEE, 2020, pp. 5145–5152.

[44] J. K. Scott et al., "Constrained zonotopes: A new tool for set-based estimation and fault detection," *Automatica*, vol. 69, pp. 126–136, 2016.

[45] G. M. Ziegler, *Lectures on polytopes*. Springer Science & Business Media, 2012.

[46] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of linear systems with uncertain parameters and inputs," in *Proc. of the 46th Conference on Decision and Control*. IEEE, 2007, pp. 726–732.

[47] X. Yang and J. K. Scott, "A comparison of zonotope order reduction techniques," *Automatica*, vol. 95, pp. 378–384, 2016.

[48] M. Wetzlinger, A. Kulmburg, and M. Althoff, "Adaptive parameter tuning for reachability analysis of nonlinear systems," in *Proc. of the 24th International Conference on Hybrid Systems: Computation and Control*. ACM, 2021.

[49] R. Farhadsefat, J. Rohn, and T. Lotfi, "Norms of interval matrices," Academy of Sciences of the Czech Republic, Institute of Computer Science, Tech. Rep., 2011.

[50] M. Althoff, "On computing the Minkowski difference of zonotopes," *arXiv preprint arXiv:1512:02794v3*, 2022.

[51] R. Rockafellar, *Convex analysis*. Princeton university press, 1972, vol. 2.

[52] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proc. of the 16th annual ACM symposium on Theory of computing*, 1984, pp. 302–311.

[53] M. Althoff et al., "ARCH-COMP21 category report: continuous and hybrid systems with linear continuous dynamics," in *Proc. of the 8th International Workshop on Applied Verification of Continuous and Hybrid Systems*, 2021, pp. 1–31.

[54] N. Kochdumper, P. Gassert, and M. Althoff, "Verification of collision avoidance for CommonRoad traffic scenarios," in *Proc. of the 8th International Workshop on Applied Verification of Continuous and Hybrid Systems*, 2021, pp. 184–194.

[55] L. H. de Figueiredo and J. Stolfi, "Affine arithmetic: Concepts and applications," *Numerical Algorithms*, vol. 37, pp. 147–158, 2004.

[56] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," in *Proc. of the 47th Conference on Decision and Control*. IEEE, 2008, pp. 4042–4048.

[57] D. Li, S. Bak, and S. Bogomolov, "Reachability analysis of nonlinear systems using hybridization and dynamics scaling," in *International Conference on Formal Modeling and Analysis of Timed Systems*. Springer, 2020, pp. 265–282.

[58] M. Wetzlinger, N. Kochdumper, S. Bak, and M. Althoff, "Fully-automated verification of linear systems using reachability analysis with support functions," in *Proc. of the 26th International Conference on Hybrid Systems: Computation and Control*. ACM, 2023.

[59] H. Abbas et al., "Probabilistic temporal logic falsification of cyber-physical systems," *Transactions on Embedded Computing Systems*, vol. 12, no. 2s, 2013.

[60] L. Mathesen, G. Pedrielli, and G. Fainekos, "Efficient optimization-based falsification of cyber-physical systems with multiple conjunctive requirements," in *Proc. of the International Conference on Automation Science and Engineering*, 2021, pp. 732–737.

[61] S. Sankaranarayanan and G. Fainekos, "Falsification of temporal properties of hybrid systems using the cross-entropy method," in *Proc. of the 15th International Conference on Hybrid Systems: Computation and Control*. ACM, 2012, pp. 125–134.

[62] H. Roehm et al., "STL model checking of continuous and hybrid systems," in *Proc. of the International Symposium on Automated Technology for Verification and Analysis*, 2016, pp. 412–427.

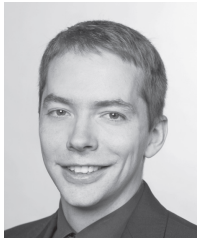
[63] O. Maler, D. Nickovic, and A. Pnueli, "From MITL to timed automata," in *Proc. of the International Conference on Formal Modeling and Analysis of Timed Systems*, 2006, pp. 274–289.

[64] G. Frehse et al., "A toolchain for verifying safety properties of hybrid automata via pattern templates," in *Proc. of the American Control Conference*, 2018, pp. 2384–2391.

Mark Wetzlinger received the B.S. degree in Engineering Sciences in 2017 jointly from Universität Salzburg, Austria and Technische Universität München, Germany, and the M.S. degree in Robotics, Cognition and Intelligence in 2019 from Technische Universität München, Germany. He is currently pursuing the Ph.D. degree in computer science at Technische Universität München, Germany. His research interests include formal verification of linear and nonlinear continuous systems, reachability analysis, adaptive parameter tuning, and model order reduction.

Niklas Kochdumper received the B.S. degree in Mechanical Engineering in 2015, the M.S. degree in Robotics, Cognition and Intelligence in 2017, and the Ph.D. degree in computer science in 2022, all from Technische Universität München, Germany. He is currently a postdoctoral researcher at Stony Brook University, USA. His research interests include formal verification of continuous and hybrid systems, reachability analysis, computational geometry, controller synthesis, and neural network verification.

Stanley Bak is an assistant professor in computer science at Stony Brook University in Stony Brook, NY, USA. He received the B.S. degree in computer science from Rensselaer Polytechnic Institute in 2007, and the M.S. degree and Ph.D. degree both in computer science from the University of Illinois at Urbana-Champaign in 2009 and 2013. His research interests include verification and testing methods for cyber-physical systems and neural networks.



Matthias Althoff is an associate professor in computer science at Technische Universität München, Germany. He received his diploma engineering degree in Mechanical Engineering in 2005, and his Ph.D. degree in Electrical Engineering in 2010, both from Technische Universität München, Germany. From 2010 to 2012 he was a postdoctoral researcher at Carnegie Mellon University, Pittsburgh, USA, and from 2012 to 2013 an assistant professor at Technische Universität Ilmenau, Germany. His research interests include formal verification of continuous and hybrid systems, reachability analysis, planning algorithms, nonlinear control, automated vehicles, and power systems.