Combining Zonotopes and Support Functions for Efficient Reachability Analysis of Linear Systems

Matthias Althoff and Goran Frehse

Abstract— **Reachability analysis is an important technique for formally verifying continuous systems, as well as for guaranteed state estimation, stability analysis, and controller synthesis. We present a detailed assessment of the computational efficiency for the reachability analysis of linear systems with respect to the two most scalable set representations: zonotopes and support functions. As a result, we propose representing reachable sets as a combination of support functions and zonotopes. This mix of representations can be converted to polyhedra of desired (directional) precision, at a higher precision compared to exclusively using support functions or zonotopes. The benefits are shown by an in-depth analysis of computational complexity and by numerical experiments.**

I. Introduction

We consider the problem of computing the set of all the states that are reachable in a dynamical system, which is also known as *set-based reachability analysis*. Despite recent advances, the trade-off between runtime and accuracy remains a central problem in reachability analysis, particularly for systems with switched dynamics.

a) Contributions: We propose a novel approach with precise control over the balance between approximation error and scalability, particularly tailored to linear systems with changing dynamics. Although successful approaches have been developed for linear systems, we revisit this problem to introduce a novel combination of zonotopes with support functions, which provides unique advantages. The findings of this paper for linear systems translate to nonlinear systems since reachability analysis for nonlinear systems often uses as an underlying technique the algorithms for linear systems, e.g., in [1]–[3]. Those approaches obtain the results for nonlinear systems by adding the linearization error as an additional uncertain input to the constantly changing underlying linear abstraction.

The main contribution of this paper is the concept of representing the reachable sets as a combination of support functions and zonotopes. This representation can be converted to constraint polyhedra of desired (directional) precision. Existing algorithms for support functions are tailored specifically to affine dynamics (linear dynamics with nondeterministic offsets). As already mentioned, nonlinear dynamics can be approximated with piecewise affine dynamics, but this requires switching the dynamics. Existing support function algorithms are not scalable when the dynamics change, as will be examined in detail in this paper.

b) Related Work: The content of this paper builds on research going back more than 20 years. Notable breakthroughs were the introduction of zonotopes for reachability analysis in [4], recurrence equations for piecewise affine dynamics in [5], the use of zonotopes for scalability of uncertain systems in [6], and the use of support functions for scalability in [7]. To the best of our knowledge, this is the first work on combining zonotopes and support functions. In this paper, we limit the discussion to switched dynamics, which change only at isolated points in time. A more general case of uncertain parameters is considered in [8] and of time-varying parameters in [9]. Other related work is cited throughout the paper.

c) Outline: In Sec. II, we briefly present zonotopes and support functions, as well as the main geometric operations on them. In Sec. III, we present the problem of approximating the flowpipes of systems with affine dynamics. In Sec. IV, we analyze and compare the complexity of implementing the algorithm from Sec. III for zonotopes and support functions. Our observations lead us to a combination of support functions and zonotopes, presented in Sec. V.

II. Efficient Set Representations for Reachability Analysis

We briefly introduce zonotopes and support functions, which are known to lead to highly scalable reachability computations. The downside of zonotopes is that they are not closed under certain set operations used in reachability analysis and the downside of support functions is that the number of evaluations may increase prohibitively if geometric operations are chained.

A. Zonotopes

A *zonotope* $\mathcal{Z} \subseteq \mathbb{R}^n$ is defined by a center $c \in \mathbb{R}^n$ and a finite number of generators $v_1, \ldots, v_k \in \mathbb{R}^n$:

$$\mathcal{Z} = \Big\{ c + \sum_{i=1}^{k} \alpha_i v_i \;\Big|\; \alpha_i \in [-1, 1] \Big\}.$$

A common denotation for zonotopes is $\mathcal{Z} = (c, \langle v_1, \ldots, v_k \rangle)$. We introduce the order of a zonotope as $o = \frac{k}{n}$. A zonotope can be seen as the Minkowski addition of line segments resulting in centrally symmetric convex polytopes as shown in Fig. 1, which illustrates how each generator spans the zonotope.

Throughout the paper we only consider the number of required binary operations (i.e. operations where two operands

This work was supported by the European Commission project UnCoVerCPS under grant number 643921.

M. Althoff is with the Department of Informatics, Technische Universität München, 85748 Garching, Germany althoff@in.tum.de

G. Frehse is with Univ. Grenoble Alpes, Laboratoire Verimag, 38400 Saint-Martin-d'Hères, France goran.frehse@imag.fr

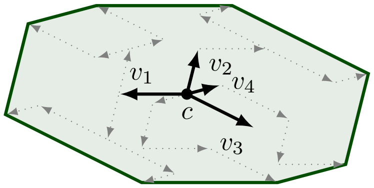


Fig. 1: A two-dimensional zonotope with center c spanned by generators v_1, v_2, v_3, v_4.

are required) for set-based operations. The only unary operations required in this paper are absolute value computation, sign changes, and concatenations of lists, whose computational effort can be safely neglected. We introduce the operator $\mathrm{Op}(\cdot)$, which returns the number of binary operations. For computational complexity (i.e. the bound on the number of binary operations in the limit), we use the well known big-O notation $\mathcal{O}(\cdot)$. Please note that we have to make certain axiomatic assumptions on the number of operations. For instance, for the multiplication Ab, where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^n$, we assume a maximum of mn multiplications and a maximum of mn additions, making $2mn$ total binary operations. Thus, we do not assume special numerical tricks that have been developed for large matrices, e.g. one can multiply two $n \times n$ matrices with complexity $\mathcal{O}\left(n^{2.376}\right)$ rather than $\mathcal{O}\left(n^3\right)$, where the latter would be considered as the "schoolbook method" [10].

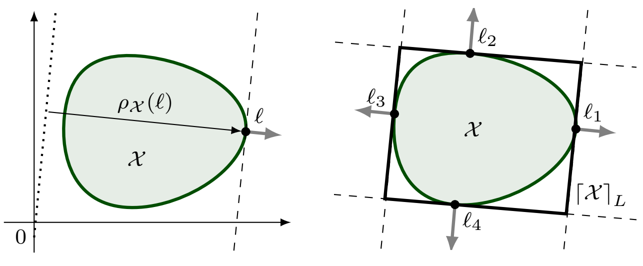
We present in Tab. I how the geometric operations required for reachability are implemented for zonotopes, which are *Minkowski sum* $\mathcal{X} \oplus \mathcal{Y} = \{x + y \mid x \in \mathcal{X}, y \in \mathcal{Y}\}$, linear maps $M\mathcal{X} = \{Mx \mid x \in \mathcal{X}\}$, and convex hull $\mathrm{CH}(\mathcal{X}, \mathcal{Y}) = \{(1-\lambda)x + \lambda y | x \in \mathcal{X},\, y \in \mathcal{Y}, \lambda \in [0, 1]\}$. Please note that the convex hull operation in Tab. I is not exact, but the error is bounded by $\sum_i \|v_i - e^{A\delta} v_i\|$ and sufficiently small for typical time step sizes in reachability analysis.

Since the number of generators of a zonotope increases during the reachability analysis due to the Minkowski sum and other operations, it is necessary to overapproximate a zonotope with another one that has fewer generators. The approach from [6] reduces the number of generators so that the resulting zonotope has a user-defined order o_R ($o_R \geq 1$). Note that this approximation is tight in the positive and negative axis directions. The computational cost for k generators is $\mathcal{O}(nk \log k)$. Zonotopes are not closed under convex hull and intersection, which motivates the introduction of support functions in the next subsection.

B. Support Functions

In this subsection, we provide definitions for polyhedra and support functions, and recall some fundamental properties. A *halfspace* $\mathcal{H} = \{x \in \mathbb{R}^n \mid a^\mathsf{T} x \leq b\}$, with normal vector $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$ is one half of the space after dividing it by a hyperplane. A *polyhedron* $\mathcal{P} \subseteq \mathbb{R}^n$ is the intersection of a finite number of halfspaces, written as

$$\mathcal{P} = \left\{ x \in \mathbb{R}^n \middle| \bigwedge\nolimits_{i=1}^{m} a_i^\mathsf{T} x \leq b_i \right\},$$



(a) Support function.

(b) Outer approximation.

Fig. 2: Evaluating the support function in a set of directions provides a polyhedral outer approximation.

where $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$. A *polytope* is a bounded polyhedron. The support function of a compact set $\mathcal{X}$ attributes to a direction $\ell \in \mathbb{R}^n$ the scalar value

$$\rho_{\mathcal{X}}(\ell) = \max\{\, \ell^\mathsf{T} x \mid x \in \mathcal{X} \}.$$

For a given direction ℓ, it defines the position of a halfspace

$$\mathcal{H}_\ell = \{x \in \mathbb{R}^n | \ell^\mathsf{T} x \leq \rho_{\mathcal{X}}(\ell)\},$$

which touches and contains $\mathcal{X}$. If ℓ is of unit length, then $\rho_{\mathcal{X}}(\ell)$ is the signed distance of $\mathcal{H}_\ell$ to the origin, see Fig. 2(a). Evaluating the support function for a set of directions $L \subseteq \mathbb{R}^n$ provides an overapproximation

$$\lceil \mathcal{X} \rceil_L = \bigcap_{\ell \in L} \{x \in \mathbb{R}^n | \ell^\mathsf{T} x \leq \rho_{\mathcal{X}}(\ell)\}, \tag{1}$$

i.e., $\mathcal{X} \subseteq \lceil \mathcal{X} \rceil_L$. If $L = \mathbb{R}^n$, then $\mathcal{X} = \lceil \mathcal{X} \rceil_L$, so the support function represents any convex set $\mathcal{X}$ exactly. If L is a finite set of directions $L = \{\ell_1, \ldots, \ell_m\}$, then $\lceil \mathcal{X} \rceil_L$ is a polyhedron, as shown in Fig. 2(b). This is also referred to as a *template polyhedron* with L being the *template directions*. The difference between using support functions and traditional methods for template polyhedra, e.g., [11], lies in the fact that the overapproximation $\lceil \mathcal{X} \rceil_L$ can be refined at any time, and incrementally, by adding more directions to L. One can interpret evaluating support functions as the lazy, on-demand, construction of a template polyhedron. The operations for support functions required by our reachability algorithm are listed in Tab. II, where we measure the number of binary operations per template direction. The final cost will depend on how many directions

TABLE I: Required operations for zonotopes, see [6].

operands	
$\mathcal{Z}_1 = (c, \langle v_1, \ldots, v_k \rangle), \mathcal{Z}_2 = (d, \langle w_1, \ldots, w_m \rangle) \subset \mathbb{R}^n, M \in \mathbb{R}^{m\times n}$	
realization	$\mathrm{Op}(\cdot)$
$\mathcal{Z}_1 \oplus \mathcal{Z}_2 = (c + d, \langle v_1, \ldots, v_k, w_1, \ldots, w_m \rangle)$	n
$M\mathcal{Z}_1 = (Mc, \langle Mv_1, \ldots, Mv_k \rangle)$	$2mn(k+1)$
$\mathrm{CH}(\mathcal{Z}_1, e^{A\delta}\mathcal{Z}_1) \subseteq \frac{1}{2}(c + e^{A\delta}c, \langle v_1 + e^{A\delta}v_1, \ldots,$ $v_k + e^{A\delta}v_k, v_1 - e^{A\delta}v_1, \ldots, v_k - e^{A\delta}v_k, c - e^{A\delta}c \rangle)$	$2n^2(k+1)+$ $2n(k+2)$

TABLE II: Required operations for support functions, see [13].

realization	$\mathrm{Op}(\cdot)$ for one direction ℓ
$\rho_{\mathcal{X}\oplus\mathcal{Y}}(\ell) = \rho_{\mathcal{X}}(\ell) + \rho_{\mathcal{Y}}(\ell), \quad (\mathcal{X},\mathcal{Y} \subseteq \mathbb{R}^n)$	1
$\rho_{M\mathcal{X}}(\ell) = \rho_{\mathcal{X}}(M^{\mathsf{T}}\ell), \quad (M \in \mathbb{R}^{m\times n})$	$2mn$
$\rho_{\mathrm{CH}(\mathcal{X},\mathcal{Y})}(\ell) = \max\{\rho_{\mathcal{X}}(\ell), \rho_{\mathcal{Y}}(\ell)\}$	1

are evaluated. Evidently, support functions do not enable us to avoid the curse of dimensionality in general: An n-dimensional approximation with a distance of ε to the real set requires $\mathcal{O}\left(1/\varepsilon^{n-1}\right)$ evaluations of the support function [12]. To combine zonotopes with support function, we require the support function of a zonotope.

Lemma 1 (Support function of a zonotope): The support function of a zonotope $\mathcal{Z} = (c, \langle v_1, \dots, v_k\rangle)$ is

$$\rho_{\mathcal{Z}}(d) = d^{\mathsf{T}}c + \sum_{i=1}^{k} |d^{\mathsf{T}}v_i|,$$

which involves $2n(k+1)$ numeric operations. □

Proof: The support function of a zonotope is [14, Prop. 2.2]

$$\rho_{\mathcal{Z}}(d) = \|M^{\mathsf{T}}d\|_1 + d^{\mathsf{T}}c = d^{\mathsf{T}}c + \sum_{i=1}^{k} |d^{\mathsf{T}}v_i|.$$

$\sum_{i=1}^{k}\sigma_i$ involves $k\,\mathrm{Op}(\sigma_i)$ operations, where $\sigma_i = |d^{\mathsf{T}}v_i|$ and $\mathrm{Op}(\sigma_i) = 2n$, resulting in $2nk + 2n = 2n(k+1)$ binary operations. □

III. Flowpipe Approximation

In our work, we consider linear systems of the form

$$\dot{x}(t) = Ax(t) + u(t), \qquad x(0) \in \mathcal{X}_0, \forall t : u(t) \in \mathcal{U}, \quad (2)$$

where $A \in \mathbb{R}^{n\times n}$ and the *input set* $\mathcal{U}$ is convex. To simplify notation, we assume that constants and input mappings are modeled within $\mathcal{U}$, e.g., dynamics of the form $\dot{x}(t) = Ax(t) + b + Bv(t)$, $\forall t : v(t) \in \mathcal{V}$ are modeled with (2) by setting $\mathcal{U} = \{b\} \oplus B\mathcal{V}$. The input signal $u(t)$ is allowed to be piecewise continuous. A trajectory $x_{x_0,u(\cdot)}(t)$ from a state x_0 is the solution of the differential equation (2) for a given initial condition $x(0) = x_0$ and a given input trajectory $u(\cdot)$, whereas $u(t)$ only refers to a value at time t:

$$x_{x_0,u(\cdot)}(t) = e^{At}x_0 + \int_0^t e^{A(t-s)}u(s)ds. \quad (3)$$

It consists of the superposition of the *homogeneous solution* ($u(\cdot) = 0$) and the *input solution* ($x_0 = 0$). Our goal is to compute the reachable set

$$\mathcal{X}_t = \{x_{x_0,u(\cdot)}(t) \mid x(0) \in \mathcal{X}_0, \forall t : u(t) \in \mathcal{U}\}.$$

For linear systems, the reachable set is (see [7], [15])

$$\begin{aligned}\mathcal{X}_t =& e^{At}\mathcal{X}_0 \oplus \int_0^t e^{A(t-s)}\mathcal{U}ds = e^{At}\mathcal{X}_0 \oplus \int_0^t e^{As}\mathcal{U}ds \\ =& e^{At}\mathcal{X}_0 \oplus \underbrace{\lim_{\delta\to 0} \bigoplus_{j=0}^{\lfloor t/\delta\rfloor} e^{A\delta j}\delta\mathcal{U}}_{=:\mathcal{Y}_t(A,\mathcal{U})}. \quad (4)\end{aligned}$$

The matrix $e^{A\delta j}$ in (4) is different for each j, so that infinitely many Minkowski sums have to be computed as $\delta \to 0$. Different methods for overapproximating the input solution $\mathcal{Y}_t$ by a finite sum have been proposed [7], [9], [14]–[16], which in essence only require Minkowski sum and linear transformation. Comparing the advantages and disadvantages of the previously developed approaches is outside the scope of this paper. Since the essence of the previously developed algorithms is the combined use of Minkowski sums and linear transformations. We use the approach in [7] (not the most accurate one), since it can be compactly written as

$$\hat{\mathcal{Y}}_{\delta N}(A,\mathcal{U}) := \bigoplus_{j=0}^{N-1} e^{A\delta j}\big(\delta\,\mathcal{U} \oplus \mathcal{E}_\delta(A,\mathcal{U})\big), \quad (5)$$

where the derivation of the error $\mathcal{E}_\delta$ is detailed in [7]. Note that we use the convention $\bigoplus_{j=0}^{-1}(\cdot) = 0$.

In this work we refer to a *flowpipe segment* when we particularly want to stress that we mean a reachable set for a time interval, which we denote as

$$\mathcal{X}_{t,t+\delta} = \bigcup_{t\le\tau\le t+\delta} \mathcal{X}_\tau.$$

A flowpipe segment over the time interval $[t, t+\delta]$ can be constructed from a flowpipe segment over $[0,\delta]$ using

$$\mathcal{X}_{t,t+\delta} = e^{At}\mathcal{X}_{0,\delta} \oplus \mathcal{Y}_t(A,\mathcal{U}). \quad (6)$$

We approximate $\mathcal{X}_{0,\delta}$ with a set $\Omega_0(X_0, U)$, whose construction will be discussed in detail in Section IV. Substituting this set in (6) and considering time points $t = \delta j$, we obtain an approximation

$$\Omega_j := e^{A\delta j}\Omega_0(X_0, U) \oplus \hat{\mathcal{Y}}_{\delta j}(A,\mathcal{U}) \quad (7)$$

as illustrated in Fig. 3. A flowpipe segment $\mathcal{X}_{0,(N+1)\delta}$ is then approximated by the union of $\Omega_0, \dots, \Omega_N$. Thus, we can represent the reachable sets for infinitely many points in time by finitely many flowpipe segments. In the next section, the above principles are assessed in terms of representing the sets via zonotopes and support functions.

IV. Complexity Assessment for Flowpipe Approximations

In this section, we compare zonotopes and support functions in terms of the runtime cost for approximating flowpipes. We do not compare the results with polyhedra since they are infeasible for larger systems due to the computational complexity of convex hulls and Minkowski addition [17] and further since the representation of results would be infeasible (a zonotope with k generators has $2k!/((k-$

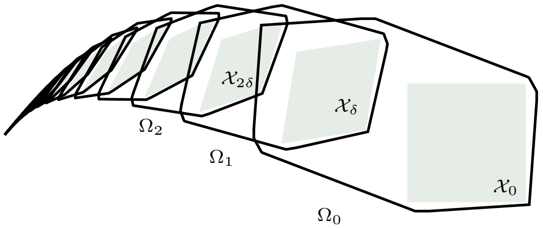


Fig. 3: A reach set cover $\Omega_0, \Omega_1, \dots$ computed with zonotopes using the implementation in [8] (solid).

$n+1)!(n-1)!)$ halfspaces [18]). The results are used to optimally combine set representations in the next section.

A. Approximating the Input Solution

We discuss different ways to compute the set of input solutions $\hat{\mathcal{Y}}_{\delta N}$ in (5). Since zonotopes are closed under linear map and Minkowski sum, and both can be computed very efficiently, they are well suited for computing $\hat{\mathcal{Y}}_{\delta N}$. Typically, $\mathcal{U}$ is a zonotope, e.g., a hyperbox. The set $\mathcal{E}_\delta$ in (5) is a hyperbox with n generators, when using the infinity norm for its computation [7].

Lemma 2 (Operations for $\hat{\mathcal{Y}}_{\delta N}$ using zonotopes): Assuming $\mathcal{U}$ is a zonotope with order o_U and $e^{A\delta}$ is known, $\mathrm{Op}(\hat{\mathcal{Y}}_{\delta N})$ is $N\left((2o_U+4)n^3+4n^2+n\right)-2n^3$. □

Proof: The matrices $e^{A\delta j}$ in (5) can be computed using the sequence $M_0 = I$, $M_{j+1} = e^{A\delta}M_j$ with $(N-1)\,2n^3$ operations, assuming that $e^{A\delta}$ is known. Each term $\mathcal{Z}_j^* = e^{A\delta j}(\delta\mathcal{U}\oplus\mathcal{E}_\delta)$ is computed with $2n^2(o_U\,n+n+2)$ operations (see Tab. I). The summation $\bigoplus_{j=0}^{N-1}\mathcal{Z}_j^*$ consists of $N\,n$ binary operations (see Tab. I) and computing N times $\mathcal{Z}_j^*$ ($N\mathrm{Op}(\mathcal{Z}_j^*)$ operations). Thus, in total, we have

$$\begin{aligned}\mathrm{Op}(\hat{\mathcal{Y}}_{\delta N}) =& N\mathrm{Op}(\mathcal{Z}_j^*) + N\,n + \mathrm{Op}(e^{A\delta j})\\ =& N\,2n^2(o_U\,n+n+2)+N\,n+(N-1)\,2n^3\\ =& N\left((2o_U+4)n^3+4n^2+n\right)-2n^3. \qquad\square\end{aligned}$$

The next lemma provides the number of required binary operations for computing $\hat{\mathcal{Y}}_{\delta N}$ when using support functions. For convenience, we denote the number of operations required to obtain a support function of $\mathcal{X}$ with

$$\mathcal{L}(\mathcal{X}) = \mathrm{Op}(\rho_{\mathcal{X}}(\ell)).$$

Lemma 3 (Operations for $\hat{\mathcal{Y}}_{\delta N}$ using supp. functions): Assuming $\mathcal{U}$ is a zonotope with order o_U and $e^{A\delta}$ is known, $\mathrm{Op}(\rho_{\hat{\mathcal{Y}}_{\delta N}}(\ell))$ is $N((o_U+1)2n^2+3n)$. □

Proof: Computing $\rho_{\hat{\mathcal{Y}}_{\delta N}}(\ell)$ involves $N(2n^2+\mathcal{L}(\mathcal{U})+n)$ operations [16]. If $\mathcal{U}$ is a zonotope with order o_U generators, then $\mathcal{L}(\mathcal{U}) = 2n(o_U\,n+1)$ according to Lemma 1, so that we obtain $N(2n^2+2n(o_U\,n+1)+n) = N((o_U+1)2n^2+3n)$.□

Next, we compare the results of the computation of $\hat{\mathcal{Y}}_{\delta N}$ when using zonotopes and support functions.

Theorem 1 (Comparing operations for $\hat{\mathcal{Y}}_{\delta N}$): The set of input solutions $\hat{\mathcal{Y}}_{\delta N}$ in (5) can be computed using zonotopes with less operations than a bounding box approximation of $\hat{\mathcal{Y}}_{\delta N}$ using support functions. At the same time, the computation with zonotopes is tighter since it exactly evaluates (5) rather than bounding the result by a box. □

Proof: A bounding box has $2n$ directions. Thus, when using support functions, we have to evaluate the support $2n$ times, which results in $2n\,N((o_U+1)2n^2+3n)$ operations using Lemma 3. Dividing the number of operations by the ones required for using zonotopes results in

$$\begin{aligned}\lim_{n\to\infty}\mu = \lim_{n\to\infty}&\frac{N\,n((o_U+1)4n^2+6n)}{N\left((2o_U+4)n^3+4n^2+n\right)-2n^3}\\ =&\frac{N(4o_U+4)}{\underbrace{N(2o_U+4)-2}_{>1}}. \qquad\square\end{aligned}$$

B. Approximating the Affine Solution

It remains to discuss the effects of the set representation on the affine solution (see (7) for $\hat{\mathcal{Y}}_{\delta j} = 0$). The difficulty lies in the initial set Ω_0. Most approximation models (see e.g. in [6], [15]) use an overapproximation based on

$$\Omega_0(X_0, U) = \mathrm{CH}\left(\mathcal{X}_0, e^{A\delta}\mathcal{X}_0\right)\oplus\mathcal{E}_{\mathcal{U}}\oplus\mathcal{E}_\Omega, \tag{8}$$

where $\mathcal{E}_{\mathcal{U}}$ compensates for errors due to the set of inputs $\mathcal{U}$ and $\mathcal{E}_\Omega$ compensates for errors due to the curvature of solutions since the convex hull assumes straight lines. We denote the order of $\mathcal{X}_0$ by o_X. An illustration of the zonotope approximation is shown in Fig. 3.

Lemma 4 (Affine solution with zonotopes): The number of operations to compute $\Omega_0,\dots,\Omega_N$ using zonotopes is $2n^2(o_X\,n+1)+2n(o_X\,n+4)$ for the initial set in (8) and $N\,2n^2(2o_X\,n+o_U\,n+n+2)$ for the propagation $e^{A\delta j}\Omega_0$ with $j=1,\dots,N$. □

Proof: The convex hull computation requires $2n^2(o_X\,n+1)+2n(o_X\,n+2)$ operations (see Tab. I) and Minkowski addition requires n operations (see Tab. I), which is used twice. Summing the operations yields the result of the lemma for the initial set. The propagation in (6) for $\mathcal{Y}_t = 0$ (homogeneous solution) requires to perform a linear map N times, where each map requires $2n^2(k+1)$ operations (see Tab. I) and the number of generators after the convex hull computation is $2o_X\,n+1$ (see Tab. I), resulting in $k = 2o_X\,n+o_U\,n+n+1$ for $\Omega_0(X_0,U)$ (see (8)). Thus, we have $2n^2(2o_X\,n+o_U\,n+n+2)$ operations for one propagation. □

Lemma 5 (Affine solution with support functions): The number of operations to compute $\Omega_0,\dots,\Omega_N$ using support functions is $2n^2+3$ for the initial set in (8) and

$$N(2n^2+2n(o_X\,n+o_U\,n+5/2)) \tag{9}$$

for the propagation $e^{A\delta j}\Omega_0$ with $j=1,\dots,N$. □

Proof: The reachable set of the first time interval in (8) is performed for one direction as (see Tab. II)

$$\rho_{\Omega_0}(\ell) = \max\left(\rho_{\mathcal{X}_0}(\ell), \rho_{\mathcal{X}_0}((e^{A\delta})^\mathsf{T}\ell)\right) + \rho_{\mathcal{E}_\mathcal{U}}(\ell) + \rho_{\mathcal{E}_\Omega}(\ell) \tag{10}$$

and requires 1 operation for the convex hull, 2 operations for the two Minkowski additions, and $2n^2$ operations for the linear map (see Tab. II). The propagation is obtained for support functions as $\ell_{j+1} = e^{A\delta^\mathsf{T}}\ell_j$,

$$\rho_{\Omega_j}(\ell) = \max\Big(\rho_{\mathcal{X}_0}(\ell_j), \rho_{\mathcal{X}_0}(\ell_{j+1}) + \delta\rho_U(\ell_j) + \rho_{\mathcal{E}_\Omega}(\ell_j)\Big).$$

Note that the value of $\rho_{\mathcal{X}_0}(\ell_{j+1})$ can be reused in the computation of $\rho_{\Omega_{j+1}}(\ell)$. Each iteration therefore requires computing one value of the support of $\mathcal{X}_0$, one of $\mathcal{U}$, and one of $\mathcal{E}_\Omega$. From the above equation and Tab. II we can see that computing the support of $\Omega_0, \Omega_1, \ldots, \Omega_N$ in a given direction ℓ requires $N(2n^2 + \mathcal{L}(\mathcal{X}_0) + \mathcal{L}(\mathcal{U}) + n)$ operations. If $\mathcal{X}_0$ and $\mathcal{U}$ are zonotopes with $k_X = o_X\, n$ ($o_X \geq 1$) and $k_U = o_U\, n$ ($o_U \geq 1$) generators, then $\mathcal{L}(\mathcal{X}_0) = 2n(o_X\, n + 1)$ and $\mathcal{L}(\mathcal{U}) = 2n(o_U\, n + 1)$. Per direction we get $N(2n^2 + 2n(o_X\, n + o_U\, n + 5/2))$. □

Next, we compare the results for the affine solution.

Theorem 2 (Comparing affine solutions): The sequence $\Omega_0, \Omega_1, \ldots, \Omega_N$ from (7) can be computed using zonotopes with about the same number of operations as a bounding box approximation using support functions (for $o_X \geq 1$). □

Proof: Since the first reachable set only has to be computed once, we focus on the costs for the propagation. A bounding box has $2n$ directions. Thus, when using support functions, we have to multiply the number of operations in (9) with $2n$, resulting in $2n\, N(2n^2 + 2n(o_X\, n + o_U\, n + 5/2))$ operations. Dividing the number of operations by the ones required for using zonotopes results in

$$\begin{aligned}\lim_{n\to\infty} \mu_{hom} &= \lim_{n\to\infty} \frac{2n\, N(2n^2 + 2n(o_X\, n + o_U\, n + 5/2))}{N\, 2n^2(2o_X\, n + o_U\, n + n + 2)} \\ &= \underbrace{\frac{2o_X + 2o_U + 2}{2o_X + o_U + 1}}_{>1}.\end{aligned}$$ □

The approximation quality of the zonotopes, however, depends on the quality of the zonotope approximation of $\mathcal{X}_0$, its generators, and the time step. While for the input convolution it seems that from a practical point of view, $\mathcal{U}$ can be approximated reasonably well be a zonotope, this need not be the case for the affine solution. In the case of hybrid systems, $\mathcal{X}_0$ may be the result of an intersection operation. An illustration of the approximation with support functions with varying degrees of precision is shown in Fig. 4.

V. Combining Support Functions and Zonotopes

In this section, we use the observations of the previous sections to devise algorithms that combine support functions and zonotopes. Theorem 1 shows that zonotopes are more efficient for computing the set of input solutions while providing more accurate results compared to support functions.

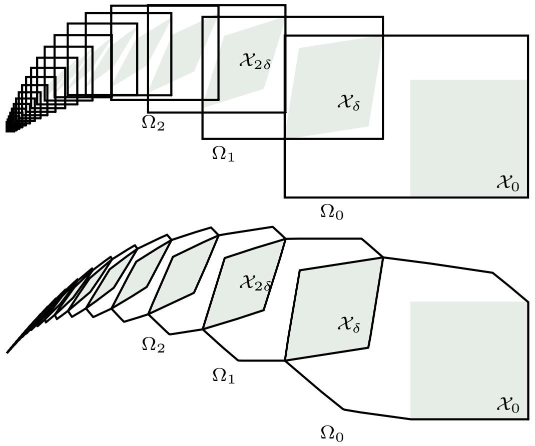


Fig. 4: A reach set cover $\Omega_0, \Omega_1, \ldots$ computed with support functions using the implementation in [19] (solid), evaluating the support function in four axis directions (top) and in 256 uniformly distributed directions (bottom).

Theorem 2 concludes that zonotopes are similarly efficient, but that the approximation quality depends very much on the time step, and whether the initial set is a zonotope. Contrary to the input set, the initial set is frequently not a zonotope, in particular if our approach is applied to hybrid systems.

Our conclusion is to use support functions for the affine solution and zonotopes for the input solution, which are combined to the overall solution:

$$\rho_{\Omega_j}(\ell) = \rho_{\Omega_0}(e^{A\delta j^\mathsf{T}}\ell) + \rho_{\hat{\mathcal{Y}}_{\delta j}}(\ell).$$

Subsequently, we examine the costs and benefits compared to the conventional support function approach. In the following, we denote the number of directions in which we evaluate support functions with $\sigma = o_\sigma\, n$ ($o_\sigma > 1$) to stress that the number of directions has to be at least linear in the dimension to ensure that the sets are closed.

A. Time-Invariant Dynamics

In the following we show that zonotopes can be used to compute, at similar costs, a support function approximation with higher precision than using support functions directly. In practice, the order of a zonotope is bounded by o_R.

Proposition 1 (Supp. function of $\hat{\mathcal{Y}}_{\delta N}$ using zonotopes): The complexity for computing the support function of $\hat{\mathcal{Y}}_{\delta N}$ for $o_\sigma\, n$ directions when it is first computed as a reduced zonotope with order o_R is

$$\mathcal{O}\left(o_U N n^3\right) + \mathcal{O}\left(o_\sigma o_R n^3\right).$$ □

Proof: According to Lemma 2, computing $\hat{\mathcal{Y}}_{\delta N}$ using zonotopes is possible with $N\left((2o_U + 4)n^3 + 4n^2 + n\right) - 2n^3$ operations. From Lemma 1 we have that constructing the support function of a zonotope for $o_\sigma\, n$ directions requires $o_\sigma\, 2n^2(k+1)$ operations, where $k = o_R\, n$, so that we have

$$\begin{aligned}&N\left((2o_U + 4)n^3 + 4n^2 + n\right) - 2n^3 + o_\sigma\, 2n^2(o_R\, n + 1) = \\ &N\left((2o_U + 4)n^3 + 4n^2 + n\right) + 2(o_\sigma o_R - 1)n^3 + 2o_\sigma n^2 = \\ &\mathcal{O}\left(o_U N n^3\right) + \mathcal{O}\left(o_\sigma o_R n^3\right).\end{aligned}$$

Since the complexity of the order reduction is only $\mathcal{O}\left(nk\log k\right)$ (see Sec. II-A), with k being the number of generators of the unreduced zonotope, the overall complexity remains $\mathcal{O}\left(o_U N n^3\right) + \mathcal{O}\left(o_\sigma o_R n^3\right)$. □

From Lemma 3 follows that computing $\hat{\mathcal{Y}}_{t_N}$ directly with support functions for $o_\sigma\, n$ directions has complexity $\mathcal{O}\left(o_\sigma o_U N n^3\right)$. Since typically $o_\sigma \gg o_U$ and $N \gg o_R$, we can conclude that first computing the input solution with zonotopes and a subsequent transformation to support functions according to Proposition 1 is beneficial in terms of computing time compared to directly computing the support function solution.

B. Time-Triggered Switching

We now turn the case of switching dynamics, where the switch is triggered at specific points in time:

$$\dot{x}(t) = A_i x(t) + u(t),\ x(0) \in \mathcal{X}_0,\ \forall t \in [\tau_i, \tau_{i+1}] : u(t) \in \mathcal{U}_i, \tag{11}$$

where A_i and $\mathcal{U}_i$ change at times τ_i. To simplify the discussion, we assume that the time between changes in the dynamics is a constant and a multiple N of the time step δ, i.e., $\tau_{i+1} - \tau_i = N\delta$. The approach is easily extended to varying N and variable time steps between switches [19].

In the following, $i \in \{1, \dots, \nu\}$ is the counter for the number of switches, ν is the total number of switches, and the index j is the time step since the last switch. We compute the reachable set at sampling times

$$t_{i,j} = \delta(iN + j).$$

For the i-th switch, we compute the transformation matrix $\hat{\Phi}_i \in \mathbb{R}^{n\times n}$, the set of input solutions $\hat{\Psi}_i \subseteq \mathbb{R}^n$, and the initial set $\Xi_i \subseteq \mathbb{R}^n$ for $\hat{\Phi}_0 = I$, $\hat{\Psi}_0 = 0$ as

$$\hat{\Phi}_{i+1} = e^{A_i \delta N} \hat{\Phi}_i, \tag{12}$$

$$\hat{\Psi}_{i+1} = e^{A_i \delta N} \hat{\Psi}_i \oplus \hat{\mathcal{Y}}_{\delta N}(A_i, U_i), \tag{13}$$

$$\Xi_i = \hat{\Phi}_i \mathcal{X}_0 \oplus \hat{\Psi}_i, \tag{14}$$

where $\hat{\mathcal{Y}}_{\delta N}(A_i, \mathcal{U}_i)$ is an overapproximation of the set of input solutions as defined in (5). Similar to (7), we obtain an overapproximation of the reachable states with

$$\Omega_{iN+j} = e^{A_i \delta j} \Omega_0(A_i, \mathcal{U}_i, \Xi_i) \oplus \hat{\mathcal{Y}}_{\delta j}(A_i, \mathcal{U}_i), \tag{15}$$

where $\mathcal{X}_{t_{i,j}, t_{i,j+1}} \subseteq \Omega_{iN+j}$. We first discuss the cost of computing the support function of (15) as before, and then investigate the benefits of using zonotopes to represent the input solution. We derive the cost of evaluating the support function of (15) analogously to Lemma 5, but with the initial set Ξ_i instead of $\mathcal{X}_0$.

Proposition 2 (Overall complexity using supp. functions): The total cost for evaluating the support function of $\Omega_0, \dots, \Omega_{iN}$ in $o_\sigma\, n$ directions is

$$\mathcal{O}\left(\nu^2 N^2 n^3 o_U o_\sigma + \nu N n^3 o_X o_\sigma\right). \qquad \square$$

Proof: After the i-th switch, computing $\rho_{\Omega_{iN}}(\ell)$,..., $\rho_{\Omega_{iN+N}}(\ell)$ in (15) involves $N(2n^2 + \mathcal{L}(\Xi_i) + \mathcal{L}(\mathcal{U}_i) + n)$ operations (see Tab. II). From the proof of Lemma 5 we have $\mathcal{L}(\mathcal{X}_0) = 2n(o_X\, n{+}1)$ and $\mathcal{L}(\mathcal{U}_i) = 2n(o_U\, n{+}1)$. From (14) and Tab. II we derive $\mathcal{L}(\Xi_i) = 2n^2 + \mathcal{L}(\mathcal{X}_0) + \mathcal{L}(\hat{\Psi}_i)$, where $\mathcal{L}(\hat{\Psi}_i)$ is obtained from Lemma 3 and (13): $\mathcal{L}(\hat{\Psi}_{i+1}) = 2n^2 + \mathcal{L}(\hat{\Psi}_i) + N((o_U + 1)2n^2 + 3n) \approx 2n^2 + \mathcal{L}(\hat{\Psi}_i) + \mathcal{O}\left(Nn^2 o_U\right)$. We simplify to $\mathcal{L}(\hat{\Psi}_i) \approx \mathcal{O}\left(iNn^2 o_U\right)$, and obtain $\mathcal{L}(\Xi_i) \approx \mathcal{O}\left(iNn^2 o_U + n^2 o_X\right)$. For the N sets after the i-th switch, we get a cost of $\mathcal{O}\left(iN^2 n^2 o_U + Nn^2 o_X\right)$. The total cost for all switches is $\mathcal{O}\left(\nu^2 N^2 n^2 o_U + \nu N n^2 o_X\right)$ and thus $\mathcal{O}\left(\nu^2 N^2 n^3 o_U o_\sigma + \nu N n^3 o_X o_\sigma\right)$ for $o_\sigma\, n$ directions.□

Corollary 1 (Complexity of input and affine solution): From Proposition 2 follows directly that the computational complexity of the input solution with respect to ν, N, and n is $\mathcal{O}\left(\nu^2 N^2 n^2\right)$ (part of the result in Proposition 2 multiplied with input order o_U) and the complexity of the affine solution with respect to ν, N, and n is $\mathcal{O}\left(\nu N n^2\right)$. In practice, N is large (often in the thousands), which makes the direct support function computation for the input solution (quadratic in ν,N) unusable, even for low-dimensional systems and a moderate number of switches. □

Next, we consider computing $\Omega_0, \dots, \Omega_{iN}$ when parts related to the input evaluation are first computed using zonotopes and then translated to support functions.

Proposition 3 (Supp. function of $\hat{\boldsymbol{\Psi}}_i$ using zonotopes): The complexity for computing the support function of $\hat{\Psi}_i$ in (13) for $o_\sigma\, n$ directions using reduced zonotopes of order o_R is $\mathcal{O}\left(\nu\, N\, n^3\right)$ for $N \gg o_R$. □

Proof: The zonotope construction of $\hat{\Psi}_{i+1}$ in (13) when $\hat{\Psi}_i$ has order o_R requires $2n^2(o_R\, n + 1)$ operations for the linear map and n operations for the Minkowski addition with $\hat{\mathcal{Y}}_{\delta N}(A_i, U_i)$ (see Tab. I) so that $\mathrm{Op}(\hat{\Psi}_{i+1}) = 2n^2(o_R\, n + 1) + n + \mathrm{Op}(\hat{\mathcal{Y}}_{\delta N})$. The order reduction of $\hat{\Psi}_{i+1}$ has complexity $\mathcal{O}\left(n\, k \log k\right)$ (see Sec. II-A) and obtaining $\hat{\mathcal{Y}}_{\delta N}$ has complexity $\mathcal{O}\left(n^3\right)$ (see Lemma 2) so that the overall complexity for obtaining the zonotope $\hat{\Psi}_{i+1}$ is $\mathcal{O}\left(n^3\right)$. Obtaining a support function has complexity $\mathcal{O}\left(n^2\right)$ according to Lemma 1 for $k = on$, resulting in $\mathcal{O}\left(n^3\right)$ for $o_\sigma\, n$ directions. Thus, the complexity for computing $\rho_{\hat{\Psi}_{i+1}}(\ell)$ remains $\mathcal{O}\left(n^3\right)$ even though the complexity of $\hat{\mathcal{Y}}_{\delta N}$ is $\mathcal{O}\left(n^3\right)$. Since we have to evaluate $\rho_{\hat{\Psi}_{i+1}}(\ell)$ for νN times, the overall complexity is $\mathcal{O}\left(\nu\, N\, n^3\right)$. □

Thus, Proposition 3 shows that the computational complexity for switched dynamics, when computing input related results with zonotopes, is linear in ν, N and cubic in n. When obtaining the same result with support functions, one has a complexity of $\mathcal{O}\left(\nu^2 N^2 n^3\right)$ according to Corollary 1.

Thus, for switched dynamics, the overall computational complexity is *quadratic* in ν and N when computing with support functions, while it is *linear* when computing $\hat{\Psi}_i$ using reduced order zonotopes.

TABLE III: Approximating the input solution $\hat{\mathcal{Y}}_{\delta N}$ of the helicopter example ($n = 28$, $N = 1000$).

method	o_σ	o_R	runtime [s]
support function (box directions)	2	–	0.42
support function (octagon directions)	56	–	12.27
support function (oct.+uniform dir.)	112	–	26.90
zonotope without reduction	–	(1000)	0.17
zonotope with reduction	–	1	0.17
zonotope with reduction	–	10	0.26
zonotope with reduction	–	20	0.35
zonotope with reduction	–	40	0.53
zonotope with reduction	–	80	0.89
zonotope with reduction	–	160	1.71
support f. (oct. dir.) from zonotope with red.	56	1	0.18
support f. (oct. dir.) from zonotope with red.	56	10	0.26
support f. (oct. dir.) from zonotope with red.	56	20	0.35
support f. (oct. dir.) from zonotope with red.	56	40	0.53
support f. (oct. dir.) from zonotope with red.	56	80	1.16
support f. (oct. dir.) from zonotope with red.	56	160	2.20

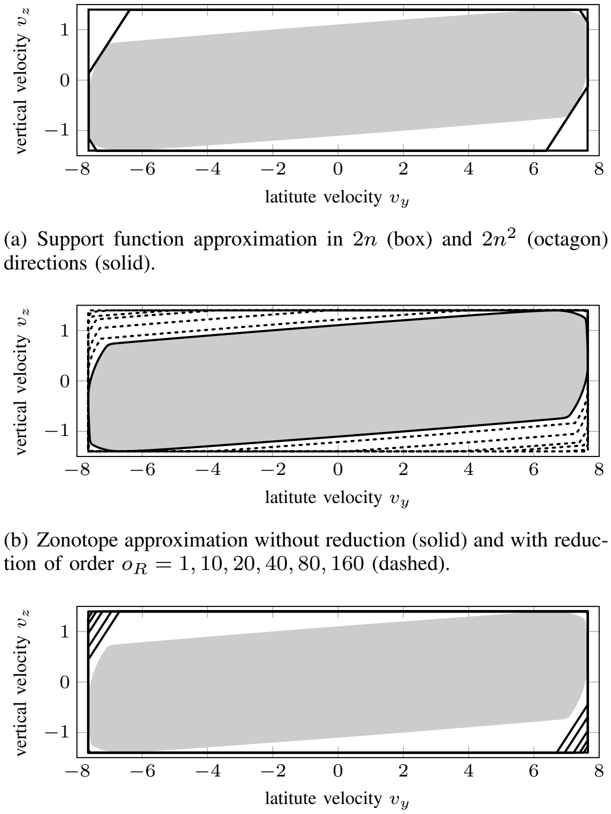
VI. Experiments

We validate the theoretical results from the previous sections on two examples, a helicopter model and a voltage converter circuit. The experiments were run on a standard MacBook laptop within a Linux Ubuntu Virtual Machine.

A. Helicopter

We compare zonotopes and support functions with a helicopter model, taken from [20]. It models the 8-dim. flight dynamics of the helicopter together with a 20-dim. H_∞-controller, so $n = 28$. The goal of the controller is to attenuate wind disturbances. We model the disturbances with $\mathcal{U}$ as $[-0.01, 0.01]$ in the dimensions that represent the rate of change in longitude, latitude, and altitude, and 0 in all other dimensions, so $\mathcal{U}$ is a zonotope of order $o_U = 3/28$. The model is stiff and all variables are highly coupled, so relatively small time steps are necessary for precise results. We compute the approximation of the input solution $\hat{\mathcal{Y}}_{\delta N}$ for $\delta = 0.003$ and $N = 1000$. The initial values of the helicopter variables in the interval $[-0.1, 0.1]$ and the controller variables are initially zero, so $\mathcal{X}_0$ is a zonotope of order $o_X = 8/28$. The runtimes for different methods are shown in Table III, and a 2-dim. projection of the obtained set is shown in Fig. 5.

The runtime of the zonotope computation without reduction is about half of the support function computation of a bounding box ($o_\sigma = 2$). This is in line with Theorem 2. The zonotope result with reduced order $o_R = 1$ is a box identical to the one produced by the support function approximation with $o_\sigma = 2$, but is computed twice as fast. Computing the support function approximation on the reduced zonotopes is considerably faster than directly computing it. Due to the choice of order reduction on zonotopes, the approximation is tight in the axis directions but not in the octagonal directions, as shown in Fig. 5(c).



(a) Support function approximation in $2n$ (box) and $2n^2$ (octagon) directions (solid).

(b) Zonotope approximation without reduction (solid) and with reduction of order $o_R = 1, 10, 20, 40, 80, 160$ (dashed).

(c) Support function approximation in $2n^2$ (octagon) directions based on reduced zonotope of order $o_R = 1, 10, 20, 40, 80, 160$ (solid).

Fig. 5: An approximation of the input solution for the helicopter example, computed using different set representations. A precise solution is shown in gray for comparison. Note that only a two-dimensional projection of the 28-dimensional approximation is shown.

B. Voltage Converter Circuit

This example models a DC-to-DC switched-mode power converter (buck-boost converter), described in [21], with continuous dynamics specified by linear ordinary differential equations. A DC-to-DC converter transforms a DC source voltage from one voltage level to another utilizing switches toggled at some (typically kilohertz) frequency with some duty cycle. The continuous dynamics of the system describe the current passing through the circuit and the output voltage, as a function of the input voltage. A clock variable is used to switch between the charging and the discharging phases in the duty cycle. We also include a global clock for tracking the startup time, so the total dimension is $n = 4$. We consider a fixed input voltage $V_s = 17.5V$ that is subjected to a continuous disturbance in the interval $[-0.1, 0.1]V$. The system is challenging despite its simple continuous dynamics, because it switches at a high frequency. The duty cycle here is $4\mu s$, while the time constant of the output voltage ranges in the milliseconds. The switched dynamics are such that overapproximation errors quickly accumulate and lead to diverging results. We use the support function implementation from [22] because it delivers higher precision

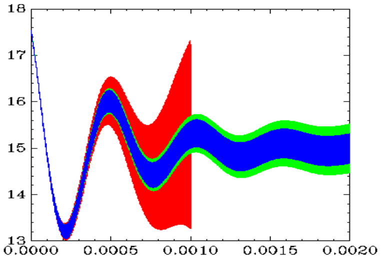


Fig. 6: Different approximations of the output voltage of a DC-to-DC converter (voltage in V over time in s). Methods 1 and 2 give nearly identical results (both dark blue). Method 3 is less precise (bright green). Method 3 with too much order reduction leads to divergence (red)

TABLE IV: Performance for the DC-to-DC converter.

method	switches ν	runtime [s]
1. support function	100	8.2
1. support function	500	248.8
1. support function	1000	1150.3
2. per-switch zonotopes	100	2.1
2. per-switch zonotopes	500	51.2
2. per-switch zonotopes	1000	269.3
3. accumulated input zonotope	100	2.2
3. accumulated input zonotope	500	23.3
3. accumulated input zonotope	1000	71.5

than the simple algorithm from Sect. III. We compare three variations. In all, the successors of initial states are support functions, while

1) the input solutions are evaluated as support functions,
2) the input solutions $\hat{\mathcal{Y}}_{\delta N}(A_i, U_i)$ are zonotopes, the accumulated input solution $\hat{\Psi}_i$ is a support function,
3) $\hat{\mathcal{Y}}_{\delta N}(A_i, U_i)$ and the accumulated input solution $\hat{\Psi}_i$ are zonotopes.

The performance results for different number of switches ν are shown in Table IV. The order of the zonotopes is reduced to $o_R = 100$. We evaluate the support function in octagonal directions plus 100 uniformly distributed directions. The results show a clear speed-up from using zonotopes, in particular for many switches.

VII. Conclusions

In the literature on reachability analysis, algorithms on zonotopes and support functions have so far been reported separately. In this paper, we compare in detail the complexity incurred by both set representations for linear dynamics. Our finding is that zonotopes are inherently suited for bounding the input solution since they provide a faster and more accurate solution. We therefore propose to use zonotopes to approximate the input solution, and the support function algorithm to compute the affine solution.

For switching dynamics at fixed points in time, the advantages for combining zonotopes with support functions are even more distinct. A direct computation with support functions would not scale even for low-dimensional systems and modest numbers of switches ($\mathcal{O}\left(\nu^2 N^2 n^3\right)$). By incorporating zonotopes, the complexity can be brought down to $\mathcal{O}\left(\nu N n^3\right)$, thus providing a practical solution that is linear in the number of switchings. The algorithm proposed in this paper lays the groundwork for bringing the power of support function algorithms to the analysis of nonlinear systems: By linearizing the system at time-triggered intervals as in [1], our algorithm can be immediately applied.

References

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

[2] E. Asarin, T. Dang, and A. Girard, "Reachability analysis of nonlinear systems using conservative approximation," in *Hybrid Systems: Computation and Control*, ser. LNCS 2623. Springer, 2003, pp. 20–35.

[3] T. Dang, O. Maler, and R. Testylier, "Accurate hybridization of nonlinear systems," in *Hybrid Systems: Computation and Control*, 2010, pp. 11–19.

[4] W. Kühn, "Rigorously computed orbits of dynamical systems without the wrapping effect," *Computing*, vol. 61, pp. 47–67, 1998.

[5] E. Asarin, T. Dang, O. Maler, and O. Bournez, "Approximate reachability analysis of piecewise-linear dynamical systems," in *HSCC'00*, ser. LNCS, 2000, pp. 20–31.

[6] A. Girard, "Reachability of uncertain linear systems using zonotopes," in *HSCC*, ser. LNCS, M. Morari and L. Thiele, Eds., vol. 3414. Springer, 2005, pp. 291–305.

[7] 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, iFAC World Congress 2008.

[8] M. Althoff, B. H. Krogh, and O. Stursberg, "Analyzing reachability of linear dynamic systems with parametric uncertainties," in *Modeling, Design, and Simulation of Systems with Uncertainties*, A. Rauh and E. Auer, Eds. Springer, 2011.

[9] M. Althoff, C. Le Guernic, and B. H. Krogh, "Reachable set computation for uncertain time-varying linear systems," in *Hybrid Systems: Computation and Control*, 2011, pp. 93–102.

[10] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *Journal of Symbolic Computation*, vol. 9, pp. 251–280, 1990.

[11] S. Sankaranarayanan, T. Dang, and F. Ivančić, "Symbolic model checking of hybrid systems using template polyhedra," in *TACAS'08*. Springer, 2008, pp. 188–202.

[12] A. V. Lotov, V. A. Bushenkov, and G. K. Kamenev, *Interactive Decision Maps*, ser. Applied Optimization. Kluwer, 2004, vol. 89.

[13] A. Girard and C. Le Guernic, "Efficient reachability analysis for linear systems using support functions," in *IFAC World Congress*, 2008.

[14] C. Le Guernic, "Reachability analysis of hybrid systems with linear continuous dynamics," Ph.D. dissertation, Université Grenoble 1 - Joseph Fourier, 2009.

[15] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Dissertation, Technische Universität München, 2010, http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20100715-963752-1-4.

[16] C. Le Guernic and A. Girard, "Reachability analysis of hybrid systems using support functions," in *CAV*, ser. LNCS, A. Bouajjani and O. Maler, Eds., vol. 5643. Springer, 2009, pp. 540–554.

[17] H. R. Tiwary, "On the hardness of computing intersection, union and Minkowski sum of polytopes," *Discrete and Computational Geometry*, vol. 40, pp. 469–479, 2008.

[18] M. Althoff, O. Stursberg, and M. Buss, "Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes," *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 233–249, 2010.

[19] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, "SpaceEx: Scalable verification of hybrid systems," in *CAV*, 2011, pp. 379–395.

[20] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*. John Wiley & Sons, 2005.

[21] L. V. Nguyen and T. T. Johnson, "Dc-to-dc switched-mode power converters," in *1st Workshop on Applied Verification for Continuous and Hybrid Systems (ARCH)*. http://cps-vo.org/node/12113, 2014.

[22] G. Frehse, R. Kateja, and C. L. Guernic, "Flowpipe approximation and clustering in space-time," in *HSCC'13*, 2013, pp. 203–212.