Reachable Set Computation for Uncertain Time-Varying Linear Systems

Matthias Althoff
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
malthoff@ece.cmu.edu

Colas Le Guernic
New York University
251 Mercer Street
New York, NY 10012
colas@cs.nyu.edu

Bruce H. Krogh
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
krogh@ece.cmu.edu

ABSTRACT

This paper presents a method for using set-based approximations to the Peano-Baker series to compute overapproximations of reachable sets for linear systems with uncertain, time-varying parameters and inputs. Alternative representations for sets of uncertain system matrices are considered, including matrix polytopes, matrix zonotopes, and interval matrices. For each representation, the computational efficiency and resulting approximation error for reachable set computations are evaluated analytically and empirically. As an application, reachable sets are computed for a truck with hybrid dynamics due to a gain-scheduled yaw controller. As an alternative to computing reachable sets for the hybrid model, for which switching introduces an additional overapproximation error, the gain-scheduled controller is approximated with uncertain time-varying parameters, which leads to more efficient and more accurate reachable set computations.

Categories and Subject Descriptors

G.1.0 [**Numerical Analysis**]: General; I.6.4 [**Simulation and Modeling**]: Model Validation and Analysis

General Terms

Algorithms, Theory, Verification

Keywords

Reachability Analysis, Linear Systems, Uncertain Parameters, Peano-Baker Series, Safety

1. INTRODUCTION

Reachable set computation offers an alternative to numerical simulation for evaluating the correctness of system models with respect to safety specifications, such as limits on system state variables. For hybrid dynamic systems, the principal difficulty is computing the reachable sets for the continuous dynamics. Research in this area is extensive, and, as summarized in the following brief literature review, a number of approaches have been developed for computing and estimating reachable states for various classes of continuous and hybrid dynamic systems. This paper presents a new method for computing overapproximations of reachable sets for linear systems with uncertain, time-varying parameters and bounded inputs using set-based approximations to the Peano-Baker series. Various methods for representing uncertain matrices can be used, allowing for a trade-off between approximation accuracy and computation time.

The method developed in this paper represents reachable sets using zonotopes, a class of polyhedra, so we focus most of our literature review on polyhedra-based methods. For so-called linear hybrid automata in which the continuous dynamics are given as constant convex polyhedral bounds on the derivative of the state vector, the reachable set can be computed directly using operations on polyhedra [13]. Reachable sets of such systems can be used as a basis for the reachability analysis of linear or even more complex systems, such as nonlinear and hybrid systems [7, 14]. The approximation of more complex continuous dynamics, and even linear dynamics, can lead to a computationally prohibitive explosion of discrete states, however. Consequently, methods that deal more directly with the continuous dynamics are often more efficient when the systems of interest cannot be modeled immediately using linear hybrid automata.

Since the set of states that can be reached over a time interval are in general nonconvex for linear dynamic systems, effective approximations of the reachable sets are constructed as unions of convex polyhedra. A number of methods and tools have been developed to approximate reachable sets for continuous dynamic systems [4, 5]. In all of these methods, a sequence of approximations to the reachable sets at discrete points in time (which *are* convex if the set of initial states and bounds on the inputs are convex) are computed and then consecutive sets in the sequence are wrapped with an enclosing convex polyhedron that accounts for the state trajectories between the discrete points in time. When the systems have bounded input signals, the approximation error can grow significantly over time due to the compounded accumulation of overapproximation added at each step to account for the effect of the inputs. In [12] it has been shown that the reachable set of linear time invariant (LTI) systems can be computed without this wrapping effect. This wrapping-free method can be implemented using different set representations, such as ellipsoids [18], polytopes [5], oriented rectangular hulls [26], zonotopes [9], or

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
HSCC'11, April 12–14, 2011, Chicago, Illinois, USA.
Copyright 2011 ACM 978-1-4503-0629-4/11/04 ...$10.00.

support functions [10], and is generally superior to methods based on approximating the system dynamics with derivative bounds. Level-set methods [27] are often more accurate than the method developed in [12], but level set methods do not scale to systems with more than a few continuous state variables (typically about four).

If uncertain parameters are considered, most algorithms work with interval methods and multidimensional intervals (axis-oriented boxes) to represent reachable sets [15, 23, 24]. Interval methods are also applied in [21] for the rigorous numerical solution of initial value problems for ordinary differential equations. The algorithms in [24] take advantage of special properties of the system dynamics, such as monotonicity. Interval methods are generally more conservative than other set representations, however. In [1], reachable sets for linear systems with uncertain time-invariant parameters are computed using zonotopes, which can be much more accurate than multidimensional intervals. The procedure presented in this paper to compute reachable sets of linear systems with time-varying parameters is a non-trivial extension of [1], since the state-transition matrix is no longer the matrix exponential for the time-varying case.

Reachability computations for linear systems with uncertain parameters can be applied to analyze nonlinear systems [2] and hybrid systems [3, 11]. Thus, the reachability analysis of linear systems can be seen as a basic module for the reachability analysis of more complicated system classes. In this paper, we illustrate how linear systems with time-varying parameters can be used to compute reachable sets for nonlinear systems. In addition, we show how time-varying parameters can be used to eliminate discrete transitions for some classes of hybrid systems, a procedure we call *continuization*, which makes it possible to compute reachable sets without the errors introduced by set intersections required to deal with transition guards in hybrid system models.

The following section reviews the overall approach to computing reachable sets for continuous dynamic systems and formulates the specific problem addressed. Section 3 develops the overapproximation of the state transition matrix based on overapproximations of matrix operations. This result is used to compute the reachable sets for systems without inputs and with bounded inputs in Sec. 4. Then we show how to perform computations with sets of matrices, using different representations. Section 5 describes matrix-matrix operations and Sec. 6 focuses on matrix-vector operations. The usefulness, efficiency, and accuracy of the proposed overapproximations are illustrated with numerical examples in Sec. 7.

2. PROBLEM FORMULATION

The basic principle of many reachability algorithms, including the approach in this paper, is to compute the reachable set for consecutive time intervals $R([t_{k-1}, t_k])$, where $t_k = k \cdot r$, $r \in \mathbb{R}^+$ is the time increment, and $k \in \mathbb{N}$ is the time step; see [5,6,9,26]. The final reachable set is then given by $R([0, t_f]) = \bigcup_{k=1}^{t_f/r} R([t_{k-1}, t_k])$, where t_f is a multiple of r. Since this finite union can be represented as an enumeration, this paper focuses on the computation of the reachable set for a single time interval $[0, r]$. The basic steps for the computation of $R([0, r])$, shown in Fig. 1, are summarized as follows.

1. Compute the reachable set at $t = r$, neglecting the input (the homogeneous solution, $R^h(r)$);
2. Generate the convex hull of the solution at $t = r$ and the initial set; and
3. Enlarge the convex hull to ensure enclosure of all trajectories for the time interval $t \in [0, r]$, including the effects of inputs.

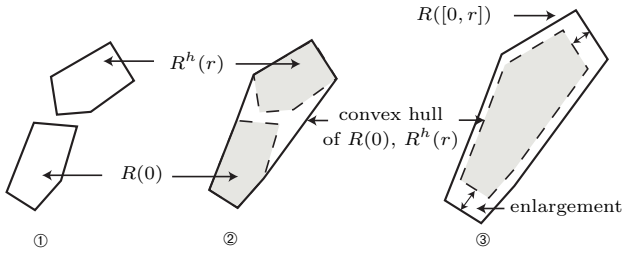


Figure 1: Steps in the computation of an overapproximation of the reachable set for a given time interval.

In this paper we consider time-varying linear systems of the form

$$\dot{x}(t) = A(t)x(t) + u(t), \quad x(0) \in \mathbb{R}^n,\ t \in [0, r], \qquad (1)$$

where for given sets $\mathcal{A} \subset \mathbb{R}^{n\times n}$ and $\mathcal{U} \subset \mathbb{R}^n$, $A : \mathbb{R}^+ \to \mathcal{A}$ and $u : \mathbb{R}^+ \to \mathcal{U}$ are piecewise continuous.

Note that the commonly used input formulation $B(t)\,\tilde{u}(t)$ with $\tilde{u}(t) \in \tilde{\mathcal{U}}$ and $B(t) \in \mathcal{B}$ is accommodated by defining $\mathcal{U} = \{B\tilde{u} | \tilde{u} \in \tilde{\mathcal{U}}, B \in \mathcal{B}\}$.

Let $\chi(t; x_0, A(\cdot), u(\cdot))$, $t \in [0, r]$, denote the solution to (1) for given $x(0) = x_0$, $A(\cdot)$, and $u(\cdot)$. Given a set of initial states, $R(0) \subset \mathbb{R}^n$, our objective is to compute the set of reachable states

$$R^e([0,r]) = \Big\{ \chi\left(t; x_0, A(\cdot), u(\cdot)\right) \Big| x_0 \in R(0), t \in [0,r], \\ \forall \tau \in [0,t] A(\tau) \in \mathcal{A}, u(\tau) \in \mathcal{U} \Big\}.$$

The superscript e on $R^e([0, r])$ denotes the exact reachable set. The exact reachable set for time-varying linear systems cannot be computed exactly [19]. Therefore, we aim to compute overapproximations $R([0, r]) \supseteq R^e([0, r])$ that are as accurate as possible, while at the same time ensuring that the computations are efficient and scale well with the system dimension n.

3. OVERAPPROXIMATING THE STATE TRANSITION MATRIX

We first consider the case when there is no input signal in (1). In this case, when $A(\cdot)$ is known, the solution is given by

$$x(t) = \Phi(t, 0)x_0,$$

where the *state transition matrix* $\Phi(t, 0)$ can be computed by the Peano-Baker series

$$\Phi(t,0) = I + \int_0^t A(\sigma_1)d\sigma_1 + \int_0^t A(\sigma_1) \int_0^{\sigma_1} A(\sigma_2)\, d\sigma_2\, d\sigma_1 \\ + \int_0^t A(\sigma_1) \int_0^{\sigma_1} A(\sigma_2) \int_0^{\sigma_2} A(\sigma_3)\, d\sigma_3\, d\sigma_2\, d\sigma_1 + \ldots \qquad (2)$$

This series converges uniformly and absolutely [25, Chap. 3]. When the system matrix is time invariant $A(\cdot) = A$, the solution of $\Phi(t, 0)$ is the well-known matrix exponential e^{At}. Since we are only interested in solutions starting at $t = 0$, we write $\Phi(t)$ as a short form of $\Phi(t, 0)$ from now on.

To obtain a numerical solution to the Peano-Baker series, the integrals in (2) can be approximated by Riemann sums of the form $\int_0^t A(\sigma_i)d\sigma_i \approx \sum_{l_i=1}^{k} A(l_i\Delta)\Delta$, where $\Delta \in \mathbb{R}^+$ is a fixed step size. For conciseness throughout the remainder of the paper, $A(m)$ denotes $A(m\Delta)$ for any integer m. Applying the Riemann sum approximation to (2) yields

$$\tilde{\Phi}(t,\Delta) := I + \sum_{l_1=1}^{k} A(l_1)\Delta + \sum_{l_2=1}^{k}\sum_{l_1=1}^{l_2} A(l_2)A(l_1)\Delta\,\Delta + \sum_{l_3=1}^{k}\sum_{l_2=1}^{l_3}\sum_{l_1=1}^{l_2} A(l_3)A(l_2)A(l_1)\Delta\Delta\Delta + \dots$$

Therefore, the matrix $\Phi(t)$ can be approximated iteratively by $\tilde{\Phi}_i(t, \Delta)$, where

$$\tilde{\Phi}_1(t,\Delta) = I + \sum_{l_1=1}^{k} A(l_1)\Delta,$$

$$\tilde{\Phi}_i(t,\Delta) = \tilde{\Phi}_{i-1}(t,\Delta) + \underbrace{\sum_{l_i=1}^{k}\dots\sum_{l_1=1}^{l_2}\left(\prod_{q=1}^{i} A(l_q)\right)\Delta^i}_{\triangleq\tilde{\Phi}_i^{\delta}(t,\Delta)}. \quad (3)$$

We now consider the computation of *sets* that overapproximate the range of state transition matrices that result when $A(t) \in \mathcal{A}$. Towards this end, we introduce the following notation and matrix operators. For a set of matrices $\mathcal{A}$, $\mathtt{CH}(\mathcal{A})$ denotes the closed convex hull of $\mathcal{A}$. Given two sets of matrices, $\mathcal{A}$ and $\mathcal{B}$, we denote by $\mathcal{A}\oplus\mathcal{B}$ and $\mathcal{A}\otimes\mathcal{B}$ the sets resulting from the point-wise sums and products of elements of $\mathcal{A}$ and $\mathcal{B}$, respectively; that is,

$$\mathcal{A}\oplus\mathcal{B} = \{A + B \mid A \in \mathcal{A},\, B \in \mathcal{B}\}, \text{ and}$$
$$\mathcal{A}\otimes\mathcal{B} = \{A \times B \mid A \in \mathcal{A},\, B \in \mathcal{B}\}$$

The point-wise product also holds for the special case when A or B is a scalar. By an abuse of notation, $\otimes$ is sometimes omitted. Considering equation (3), we only know that every $A(l_q)$ belongs to some set $\mathcal{A}$. Thus, $\prod_{q=1}^{m} A(l_q)$ belongs to $\bigotimes_{q=1}^{m}\mathcal{A}$, which we will denote by $\mathcal{A}^m$ in the following,[1] and

$$\tilde{\Phi}_i^{\delta}(t,\Delta) \in \bigoplus_{l_i=1}^{k}\dots\bigoplus_{l_1=1}^{l_2}\mathcal{A}^i\Delta^i. \quad (4)$$

The following proposition on distributivity of multiplication by positive scalars over addition for convex matrix sets is useful to simplify the above expression,

PROPOSITION 1. (Distributivity of Matrix Sets) *If $\mathcal{A}$ is convex and $a, b \in \mathbb{R}^+$, then*

$$a\mathcal{A}\oplus b\mathcal{A} = (a+b)\mathcal{A}.$$

PROOF. It is always true that $(a+b)\mathcal{A} \subseteq a\mathcal{A}\oplus b\mathcal{A}$, even if $\mathcal{A}$ is not convex. Further, due to the convexity it follows for $X_1, X_2 \in \mathcal{A}$ and $\alpha \in [0,1]$ that $\alpha X_1 + (1-\alpha)X_2 \in \mathcal{A}$. Making use of $a, b \geq 0$, let $\alpha = \frac{a}{a+b}$, which gives

$$\frac{a}{a+b}X_1 + \frac{b}{a+b}X_2 \in \mathcal{A}.$$

Thus, $aX_1 + bX_2 \in (a+b)\mathcal{A}$ and consequently $a\mathcal{A}\oplus b\mathcal{A} \subseteq (a+b)\mathcal{A}$. □

Using this proposition, the following theorem provides an expression for an overapproximation for the set of possible state transition matrices $\Phi(t)$.

THEOREM 1. (Set of State Transition Matrices) *Let $\mathcal{M}(t)$ denote the set of state transition matrices $\Phi(t)$ when $A(\tau) \in \mathcal{A}$ for $\tau \in [0,t]$. Then $\mathcal{M}(t) \subseteq \overline{\mathcal{M}}(t)$, where*

$$\overline{\mathcal{M}}(t) = \bigoplus_{i=0}^{\infty}\overline{\mathcal{M}}_i(t), \qquad \overline{\mathcal{M}}_i(t) = \frac{t^i}{i!}\mathtt{CH}(\mathcal{A}^i).$$

PROOF. Proposition 1 implies the set in equation (4) is contained in $\Delta^i\left(\sum_{l_i=1}^{k}\dots\sum_{l_1=1}^{l_2}1\right)\mathtt{CH}(\mathcal{A}^i)$. The summation of ones is computed using the formula

$$\sum_{l=1}^{k} l^m = \frac{k^{m+1}}{m+1} + \mathcal{O}(k^m)$$

from which the auxiliary results ξ_m are obtained:

$$\xi_1 = \sum_{l_1=1}^{l_2} 1 = l_2 + \mathcal{O}(l_2^0)$$

$$\xi_2 = \sum_{l_2=1}^{l_3}\xi_1 = \sum_{l_2=1}^{l_3} l_2 + \mathcal{O}(l_2^0) = \frac{l_3^2}{2} + \mathcal{O}(l_3^1)$$

$$\dots$$

$$\xi_i = \sum_{l_i=1}^{k}\dots\sum_{l_1=1}^{l_2} 1 = \frac{k^i}{i!} + \mathcal{O}(k^{i-1}).$$

From the relation $k\Delta = t$ it follows that $\tilde{\Phi}_i^{\delta}(t,\Delta)$ is in $\left(\frac{t^i}{i!} + \Delta\mathcal{O}(t^{i-1})\right)\mathtt{CH}(\mathcal{A}^i)$. Taking the limit as $\Delta \to 0$, the Riemann sums used to approximate the integrals in (2) converge and therefore

$$\Phi_i^{\delta}(t) \subseteq \frac{t^i}{i!}\mathtt{CH}(\mathcal{A}^i). \quad \square$$

Note that the computation of $\overline{\mathcal{M}}(t)$ resembles the computation of the set of Taylor expansions of e^{At}, namely, $\{\sum_{i=0}^{\infty} A^i t^i/i! | A \in \mathcal{A}\}$, except the relationships between the different occurrences of A are forgotten in Theorem 1 and the convex hull of $\mathcal{A}^i$ has to be computed. We will see in Sec. 5 that this computation requires no additional work in our multiplication procedure since we only represent convex sets.[2].

Defining the norm of a set of matrices $\mathcal{M}$ as

$$\|\mathcal{M}\| = \sup\{\|M\| \,|\, M \in \mathcal{M}\},$$

[1] Note that $\mathcal{A}^m$ is $\{\prod_{q=1}^{m} A_i \mid A_i \in \mathcal{A}\}$, which is different from and larger than $\{A^m \mid A \in \mathcal{A}\}$.

[2] Even if $\mathcal{A}$ is convex, $\mathcal{A}^2$ might not be, but our approximate multiplication gives a convex overapproximation of $\mathcal{A}^2$, and thus of $\mathtt{CH}(\mathcal{A}^2)$.

where $\|M\|$ denotes a particular matrix norm, the norm of $\overline{\mathcal{M}}(t)$ is bounded as

$$\|\overline{\mathcal{M}}(t)\| \leq 1 + \|\mathcal{A}\|t + \frac{1}{2!}\|\mathcal{A}\|^2t^2 + \frac{1}{3!}\|\mathcal{A}\|^3t^3 + \ldots = e^{\|\mathcal{A}\|t}$$

which implies that $\|\overline{\mathcal{M}}(t)\| < \infty$ for $\|\mathcal{A}\| < \infty$ and $t < \infty$.

To approximate $\overline{\mathcal{M}}(t)$, the following proposition shows that we can replace the infinite sum $\overline{\mathcal{M}}(t) = \sum_{i=0}^{\infty}\overline{\mathcal{M}}_i(t)$ with a finite sum of η terms, $\sum_{i=0}^{\eta}\overline{\mathcal{M}}_i(t)$, plus a set that bounds the remaining terms in the infinite sum. For a set of matrices $\mathcal{M}$, the notation $|\mathcal{M}|$ denotes the matrix in which each element is equal to the supremum of the absolute value of the corresponding element in each matrix in $\mathcal{M}$. That is, $|\mathcal{M}|(i,j) = \sup\{|M(i,j)| \,|\, M \in \mathcal{M}\}$.

PROPOSITION 2. (State Transition Remainder) *The set of remainder matrices $\mathcal{E}(t)$ is an overapproximation of $\bigoplus_{i=\eta+1}^{\infty}\overline{\mathcal{M}}_i(t)$ computed as*

$$\mathcal{E}(t) = [-W(t), W(t)], \quad W(t) = e^{|\mathcal{A}|t} - \sum_{i=0}^{\eta}\frac{t^i}{i!}|\mathcal{A}|^i.$$

PROOF. By induction it follows that $|\mathcal{A}^n| \leq |\mathcal{A}|^n$ element wise. Thus,

$$\left|\bigoplus_{i=\eta+1}^{\infty}\frac{t^i}{i!}\mathtt{CH}(\mathcal{A}^i)\right| \leq \sum_{i=\eta+1}^{\infty}\frac{t^i}{i!}|\mathcal{A}|^i = e^{t|\mathcal{A}|} - \sum_{i=0}^{\eta}\frac{t^i}{i!}|\mathcal{A}|^i. \quad \square$$

4. OVERAPPROXIMATING THE REACHABLE SET

For autonomous uncertain time-varying systems, the set of state transition matrices makes it possible to bound the state of an autonomous system by $x(r) \in \overline{\mathcal{M}}(r)x(0)$ so that the reachable set is obtained by $R(r) = \overline{\mathcal{M}}(r)R(0)$. In this section, we first derive an overapproximation for the reachable set for a time interval, $R([0,r])$, for autonomous systems and then show how to incorporate the effects of uncertain inputs.

The set $R^e([0,r])$ can be approximated by the convex hull of $R(0)$ and $R(r)$. To ensure this approximation is an overapproximation we add an error term $\mathcal{F}(r) \otimes R(0)$ to this convex hull.

For a given trajectory starting from x_0, we know that for any t in $[0,r]$, $x(t)$ is in $\overline{\mathcal{M}}(t)x_0$. Therefore, Theorem 1 implies there exists a sequence of matrices $A_{t,i} \in \mathtt{CH}(A^i)$ such that

$$x(t) = \sum_{i=0}^{\infty}\frac{t^i}{i!}A_{t,i}x_0$$

We approximate $x(t)$ by a point $\hat{x}(t)$ in the convex hull of x_0 and $\overline{\mathcal{M}}(r)x_0$ defined as

$$\hat{x}(t) = (1 - \frac{t}{r})x_0 + \frac{t}{r}\left(\sum_{i=0}^{\infty}\frac{r^i}{i!}A_{t,i}\right)x_0.$$

Because of its dependence on t, $\hat{x}(t)$ may not describe a straight line from x_0 and $x(r)$ as t varies from 0 to r, but it will always stay in the convex hull of x_0 and $\overline{\mathcal{M}}(r)x_0$.

We now evaluate the error made when applying this approximation.

$$\begin{aligned} x(t) - \hat{x}(t) &= \sum_{i=0}^{\infty}\frac{t^i}{i!}A_{t,i}x_0 - (1-\frac{t}{r})x_0 - \frac{t}{r}\left(\sum_{i=0}^{\infty}\frac{r^i}{i!}A_{t,i}\right)x_0 \\ &= \left(\sum_{i=2}^{\infty}\frac{(t^i - tr^{i-1})}{i!}A_{t,i}\right)x_0 \end{aligned}$$

For $i > 1$, one can show [1] that

$$\left\{t^i - tr^{i-1} \mid t \in [0,r]\right\} = \left[\left(i^{\frac{-i}{i-1}} - i^{\frac{-1}{i-1}}\right)r^i, 0\right].$$

Thus, we can define $\mathcal{F}(r)$ as

$$\mathcal{F}(r) = \bigoplus_{i=2}^{\infty}\frac{r^i}{i!}\mathtt{CH}\left(\{0\} \cup \left(i^{\frac{i}{1-i}} - i^{\frac{1}{1-i}}\right)\mathcal{A}^i\right).$$

And we have $R([0,r]) \subseteq \mathtt{CH}\left(R(0) \cup \overline{\mathcal{M}}(r)R(0)\right) \oplus \mathcal{F}(r)R(0)$.

Similarly to $\overline{\mathcal{M}}(r)$, $\mathcal{F}(r)$ can be overapproximated by considering the sum up to η and overapproximating the remaining terms by $\mathcal{E}(r)$. For an efficient evaluation, $\mathcal{A}$ is overapproximated by an interval matrix (specified later) and interval arithmetic is applied to obtain an overapproximation of $\mathcal{F}(r)$.

We now consider the additional reachable set due to uncertain inputs. Since the superposition principle for linear systems can be applied, the reachable set due to the input can be computed independently of the reachable set for the autonomous system.

THEOREM 2 (INPUT SOLUTION). *The set of reachable states due to the uncertain input $u(t) \in \mathcal{U}$ can be overapproximated by*

$$\mathcal{P}(t) = \bigoplus_{i=0}^{\eta}\left(\frac{t^{i+1}}{(i+1)!}\mathtt{CH}(\mathcal{A}^i\mathcal{U})\right) \oplus \frac{t}{\eta+2}\mathcal{E}(t)\{|\mathcal{U}|\}.$$

PROOF. The differential equation $\dot{x}(t) = A(t)x(t) + u(t)$ can be rewritten as

$$\frac{d}{dt}\begin{pmatrix}x(t)\\1\end{pmatrix} = \underbrace{\begin{pmatrix}A(t) & u(t)\\0 & 0\end{pmatrix}}_{A_u(t)}\begin{pmatrix}x(t)\\1\end{pmatrix}$$

Based on Theorem 1, the set of points reachable at time t from $(x_0,1)^\top$ is included in $\bigoplus_{i=0}^{\infty}\frac{t^i}{i!}\mathtt{CH}(\mathcal{A}_u^i)\{(x_0,1)^\top\}$ where

$$\mathcal{A}_u = \left\{\begin{pmatrix}A & u\\0 & 0\end{pmatrix}\Big| A \in \mathcal{A},\, u \in \mathcal{U}\right\}.$$

One can show by induction that, for $i > 0$,

$$\mathcal{A}_u^i = \left\{\begin{pmatrix}A_{i-1}A & A_{i-1}u\\0 & 0\end{pmatrix}\Big| A \in \mathcal{A},\, A_{i-1} \in \mathcal{A}^{i-1},\, u \in \mathcal{U}\right\}.$$

Thus, taking $x_0 = (0,\ldots,0)$, we have

$$\mathcal{P}(t) = \bigoplus_{i=0}^{\infty}\frac{t^{i+1}}{(i+1)!}\mathtt{CH}(\mathcal{A}^i \otimes \mathcal{U}).$$

Similarly to Proposition 2, we can compute this infinite sum up to η and bound the remainder by

$$\begin{aligned}\left|\bigoplus_{i=\eta+1}^{\infty}\frac{t^{i+1}}{(i+1)!}\mathtt{CH}(\mathcal{A}^i \otimes \mathcal{U})\right| &\leq \frac{t}{\eta+2}\sum_{i=\eta+1}^{\infty}\left(\frac{\eta+2}{i+1}\right)\frac{t^i}{i!}|\mathcal{A}|^i|\mathcal{U}| \\ &\leq \frac{t}{\eta+2}W(t)|\mathcal{U}| \quad \square\end{aligned}$$

If the origin is contained in the set of possible inputs ($0 \in \mathcal{U}$), it holds that $\mathcal{P}([0,r]) = \mathcal{P}(r)$; see [1]. If this is not the case, some correction measures have to be applied [1]. Algorithm 1 summarizes the steps for computing $R([0,t_f])$ under the assumption $0 \in \mathcal{U}$. Note that the error of the computations in Algorithm 1 can be made arbitrarily small when $r \to 0$ while the computational effort grows. When enforcing an upper bound on the number of parameters describing the reachable set, the error cannot be made arbitrarily small since overapproximative order reduction techniques have to be applied to bound the growing number of set parameters.

Algorithm 1 Compute $R([0,t_f])$

Require: Initial set $R(0)$, set of state transition matrices $\overline{\mathcal{M}}(r)$, input set $\mathcal{U}$, set of correction matrices $\mathcal{F}$, time horizon t_f, time step r.
Ensure: $R([0,t_f])$

$\mathcal{H}_0 = \mathtt{CH}(R(0) \cup \overline{\mathcal{M}}(r)R(0)) \oplus \mathcal{F}(r)R(0)$
$\mathcal{P}_0 = \mathcal{P}(r)$
$R_0 = \mathcal{H}_0 \oplus \mathcal{P}_0$
for $k = 1 \dots t_f/r - 1$ **do**
  $R_k = \overline{\mathcal{M}}(r)R_{k-1} \oplus \mathcal{P}_0$
end for
$R([0,t_f]) = \bigcup_{k=1}^{t_f/r} R_{k-1}$

The proof of Theorem 2 uses the fact that we can express a d-dimensional system with inputs as a $(d+1)$-dimensional autonomous system. By not using this transformation, we can use different representations and algorithms for the set of matrices $\mathcal{A}$ and the set of inputs $\mathcal{U}$. On a side note, one can use a similar transformation to ensure that $0 \in \mathcal{U}$.

5. COMPUTING WITH SETS OF MATRICES

The computation of $\overline{\mathcal{M}}(t)$ requires representations of sets of matrices and methods for computing sums and products of sets of matrices using these representations. We note that the space of matrices is itself a vector space, where the inner product of two matrices A and B in $\mathbb{R}^{m\times n}$ can be defined as

$$\langle A \mid B \rangle = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}b_{ij} = trace(AB^\top),$$

A number of representations can be used to characterize sets of matrices. Here we consider polytopes in vertex representation, zonotopes, and interval products. Algorithms are known for computing the Minkowski sum for each of these representations; see, e.g., [9]. We describe here how to over-approximate the product of two sets using each representation.

5.1 Matrix Polytopes

We define a matrix polytope, designated by the superscript $[p]$, as the convex hull of a set of matrix vertices $V^{(i)}$; that is,

$$\mathcal{A}^{[p]} = \Big\{ \sum_{i=1}^{r_A} \alpha_i V^{(i)} \Big| V^{(i)} \in \mathbb{R}^{n\times n}, \alpha_i \geq 0, \sum_i \alpha_i = 1 \Big\}.$$

The multiplication of two matrix polytopes $\mathcal{A}^{[p]}\mathcal{B}^{[p]}$, where the matrix vertices of $\mathcal{B}^{[p]}$ are denoted by $W^{(i)}$, can be over-approximated by another matrix polytope $\mathcal{C}^{[p]}$ given by

$$\begin{aligned}
&\mathcal{A}^{[p]} \otimes \mathcal{B}^{[p]} \\
&= \Big\{ (\sum_{i=1}^{r_A} \alpha_i V^{(i)})(\sum_{j=1}^{r_B} \beta_j W^{(j)}) \Big| \alpha_i, \beta_j \geq 0, \sum_{i=1}^{r_A} \alpha_i = 1, \sum_{j=1}^{r_B} \beta_j = 1 \Big\} \\
&= \Big\{ \sum_{i=1}^{r_A}\sum_{j=1}^{r_B} \alpha_i\beta_j V^{(i)}W^{(j)} \Big| \alpha_i, \beta_j \geq 0, \sum_{i=1}^{r_A} \alpha_i = 1, \sum_{j=1}^{r_B} \beta_j = 1 \Big\} \\
&\subseteq \Big\{ \sum_{i=1}^{r_C} \gamma_i X^{(i)} \Big| \gamma_i \geq 0, \sum_{i=1}^{r_C} \gamma_i = 1 \Big\} = \mathcal{C}^{[p]},
\end{aligned} \tag{5}$$

where

$$\begin{array}{llll} \gamma_1 = \alpha_1\beta_1, & \gamma_2 = \alpha_1\beta_2, & \dots \\ X^{(1)} = V^{(1)}W^{(1)}, & X^{(2)} = V^{(1)}W^{(2)}, & \dots \end{array} \tag{6}$$

In order to show that $\mathcal{C}^{[p]}$ is an overapproximation, as indicated in (5), it has to be shown that for each α_i and β_j value it follows that $\gamma_i \geq 0$ and that $\sum_{i=1}^{r_C} \gamma_i = 1$, i.e. for each matrix $A \in \mathcal{A}^{[p]}$, $B \in \mathcal{B}^{[p]}$, it is true that $AB \in \mathcal{C}^{[p]}$. In (6) it can be immediately seen that the first property $\gamma_i \geq 0$ is always fulfilled. The second property is always true since

$$\sum_{i=1}^{r_C} \gamma_i = \sum_{i=1}^{r_A}\sum_{j=1}^{r_B} \alpha_i\beta_j = \underbrace{\sum_{i=1}^{r_A} \alpha_i}_{=1} \underbrace{\sum_{j=1}^{r_B} \beta_j}_{=1} = 1.$$

The reciprocal property that for each $C \in \mathcal{C}^{[p]}$, there exists matrices $A \in \mathcal{A}^{[p]}$, $B \in \mathcal{B}^{[p]}$ such that $AB = C$, is not always true, meaning the $\mathcal{C}^{[p]}$ is an overapproximation.

The disadvantage of matrix polytopes is their combinatorial complexity. For m vertices, the number of vertices of the l^{th} power is m^l. The Minkowski addition of polytopes up to order l can be done by adding each vertex of the solution up to the $(l-1)^{th}$ order with each vertex of the l^{th} power resulting in $\prod_{i=1}^{l} m^i = m^{\sum_{i=1}^{l} i} = m^{0.5(l(l+1))} = \mathcal{O}(m^{l^2})$ matrices.

5.2 Matrix Zonotopes

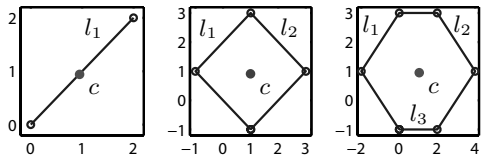
Zonotopes, which are polytopes defined as a sum of segments, are useful for computing reachable sets for high-dimensional systems; see e.g. [9]. In a vector space $\mathbb{V}$, zonotopes are specified by a center $c \in \mathbb{V}$ and generators $g^{(i)} \in \mathbb{V}$ as

$$Z = \Big\{ x = c + \sum_{i=1}^{e} \beta_i\, g^{(i)} \Big| \beta_i \in [-1,1] \Big\}. \tag{7}$$

Zonotopes are always centrally symmetric to the center c and their order is defined by $\rho = \frac{e}{n}$ (e: number of generators, n: dimension). Zonotopes are denoted in a short form as $Z = (c, g^{(1)}, \dots, g^{(e)})$. One can interpret the above definition as the Minkowski addition of line segments $l_i = \beta_i\, g^{(i)}$, $\beta_i \in [-1,1]$ as illustrated step-by-step in Fig. 2, with $\mathbb{V} = \mathbb{R}^2$.

The use of zonotopes, usually in $\mathbb{R}^n$, is justified by the fact that linear transformation and Minkowski addition, which are both of great importance for reachability analysis, can be computed efficiently; see [9, 12]. Given two zonotopes $Z_1 = (c_1, g^{(1)}, \dots, g^{(e)})$ and $Z_2 = (c_2, h^{(1)}, \dots, h^{(u)})$,

$$\begin{aligned} L\,Z_1 &= (Lc_1, Lg^{(1)}, \dots, Lg^{(e)}), \quad L \in \mathbb{R}^{n\times n} \\ Z_1 \oplus Z_2 &= (c_1 + c_2, g^{(1)}, \dots, g^{(e)}, h^{(1)}, \dots, h^{(u)}). \end{aligned} \tag{8}$$



(a) $c + l_1$ (b) $c + l_1 + l_2$ (c) $c + l_1 + l_2 + l_3$

Figure 2: Step-by-step construction of a zonotope from left to right via Minkowski addition of line segments.

The Minkowski sum of two zonotopes is simply an addition of their centers and a concatenation of their generators.

Here we are interested in zonotopes in the space of matrices. We will call them matrix zonotopes and denote them with a superscript $[z]$. A matrix zonotope is defined as

$$\mathcal{A}^{[z]} = \Big\{ G^{(0)} + \sum_{i=1}^{\kappa_A} p_i G^{(i)} \Big| p_i \in [-1, 1], G^{(i)} \in \mathbb{R}^{n \times n} \Big\},$$

and written in short form as $(G^{(0)}, G^{(1)}, \ldots, G^{(\kappa_A)})$, where the first matrix is referred to as the *matrix center* and the other matrices as *matrix generators*.

The multiplication of two matrix zonotopes $\mathcal{A}^{[z]}\mathcal{B}^{[z]}$, where the matrix generators of $\mathcal{B}^{[z]}$ are denoted by $H^{(i)}$, can be overapproximated by another matrix zonotope $\mathcal{C}^{[z]}$ given by

$$\begin{aligned}
\mathcal{A}^{[z]}\mathcal{B}^{[z]} &= \left(G^{(0)} \oplus \bigoplus_{i=1}^{\kappa_A} [-1, 1] G^{(i)} \right) \left(H^{(0)} \oplus \bigoplus_{j=1}^{\kappa_B} [-1, 1] H^{(j)} \right) \\
&\subseteq G^{(0)} H^{(0)} \oplus \bigoplus_{\substack{i=0 \\ (i,j) \neq (0,0)}}^{\kappa_A} \bigoplus_{j=0}^{\kappa_B} [-1, 1][-1, 1] G^{(i)} H^{(j)} \\
&= I^{(0)} \oplus \bigoplus_{i=1}^{\kappa_C} [-1, 1] I^{(i)} = \mathcal{C}^{[z]},
\end{aligned} \tag{9}$$

where $I^{(1)} = G^{(0)} H^{(1)}, I^{(2)} = G^{(0)} H^{(2)}, \ldots$

It can be directly seen that $\mathcal{C}^{[z]} \supseteq \mathcal{A}^{[z]}\mathcal{B}^{[z]}$. However, the resulting set of matrices is not exact.

An advantage of matrix zonotopes over matrix polytopes is that they are much more compact in their representation, a property inherited from zonotopes; see [3]. Another advantage is that the Minkowski addition of the two matrix zonotopes $\mathcal{A}^{[z]}$ and $\mathcal{B}^{[z]}$ is computationally cheap since one only has to add their matrix centers and concatenate their matrix generators. That is,

$$\mathcal{A}^{[z]} \oplus \mathcal{B}^{[z]} = (G^{(0)} + H^{(0)}, G^{(1)}, \ldots, G^{(\kappa_A)}, H^{(1)}, \ldots, H^{(\kappa_B)}).$$

The number of matrix generators for the l^{th} power is $(\kappa + 1)^l - 1$. Thus, the number of matrix generators of $\overline{\mathcal{M}}(t)$ up to order l is $\sum_{k=0}^{l} (\kappa+1)^k - 1 = (1-(\kappa+1)^{l+1})/(-\kappa) - (l+1) = \mathcal{O}(\kappa^l)$. The computational complexity can be drastically reduced by applying order reduction techniques developed for zonotopes; see [9,17]. Equally powerful reduction techniques for polytopes are not known to the authors.

5.3 Interval Matrices

An interval matrix is a special case of a matrix zonotope specified by intervals for each element; that is

$$\mathcal{A}^{[i]} = [\underline{A}, \overline{A}], \quad \forall i, j : \underline{A}_{ij} \leq \overline{A}_{ij}, \quad \underline{A}, \overline{A} \in \mathbb{R}^{n \times n}.$$

The matrix $\underline{A}$ is referred to as the *lower bound* and $\overline{A}$ as the *upper bound* of $\mathcal{A}$.

Interval matrix multiplications are performed using interval arithmetic [16]. The addition and multiplication rules for two real-valued intervals $a^I = [\underline{a}, \overline{a}]$ and $b^I = [\underline{b}, \overline{b}]$ are given by

$$\begin{aligned}
a^I + b^I &= [\underline{a} + \underline{b}, \overline{a} + \overline{b}], \\
a^I \cdot b^I &= [\min(\underline{a}\,\underline{b}, \underline{a}\,\overline{b}, \overline{a}\,\underline{b}, \overline{a}\,\overline{b}), \max(\underline{a}\,\underline{b}, \underline{a}\,\overline{b}, \overline{a}\,\underline{b}, \overline{a}\,\overline{b})].
\end{aligned} \tag{10}$$

Using the rules in (10), the multiplication of an interval matrix $\mathcal{A}^{[i]}$ with another interval matrix $\mathcal{B}^{[i]}$ is computed elementwise: $\mathcal{C}_{ij}^{[i]} = \sum_{k=1}^{n} \mathcal{A}_{ik}^{[i]} \mathcal{B}_{kj}^{[i]}$. Clearly, the result is an overapproximation. In contrast to matrix polytopes and matrix zonotopes, where the number of vertices and generators grow after addition and multiplication, the representation does not grow for interval matrices.

6. COMPUTATION OF THE REACHABLE SET

This section shows how the representations defined in the previous section can be use to compute reachable set approximations. In order to compute $\overline{\mathcal{M}}(r) R_k$ in Algorithm 1, the multiplication of a matrix zonotope/polytope or an interval matrix with a zonotope has to be computed. For a small enough time t (as it is typically the case for reachability analysis), the terms $\overline{\mathcal{M}}_i(t)$ for large i values, which are referred to as higher order terms, contribute less to the computation of $\overline{\mathcal{M}}(t)$. Thus, one should use sophisticated computations for the first terms and switch to coarser and more efficient computations for higher order terms. For instance, the first two terms of a matrix zonotope could be computed using matrix zonotope computations, while the other terms are computed via interval matrix overapproximations. Denoting the set of transition matrices using matrix zonotopes by $\overline{\mathcal{M}}^{[z]}(t)$ and the set using interval matrices by $\overline{\mathcal{M}}^{[i]}(t)$, the reachable set is computed as $R_k = \overline{\mathcal{M}}^{[z]}(r) R_{k-1} \oplus \overline{\mathcal{M}}^{[i]}(r) R_{k-1} \oplus \mathcal{P}_0$. This technique is preferred over the transformation of the interval matrix to a matrix zonotope, making it possible to obtain $\overline{\mathcal{M}}(t)$, since the transformation results in too many generators, especially in high dimensional spaces.

In order to tightly overapproximate an interval matrix multiplication, the interval matrix $\mathcal{A}^{[i]}$ is split into a real valued part $A^{[n]} \in \mathbb{R}^{n \times n}$ and a symmetric interval matrix $\mathcal{S} = [-S, S]$: $\mathcal{A}^{[i]} Z \subseteq A^{[n]} Z \oplus \mathcal{S} Z$. The following proposition shows how to compute the symmetric interval matrix part in the zonotope multiplication (8).

PROPOSITION 3 (INTERVAL MATRIX MULTIPLICATION). *The multiplication of a symmetric interval matrix $\mathcal{S} = [-S, S]$ with a zonotope $Z = (c, g^{(1)}, \ldots, g^{(e)})$ can be overapproximated by a hyperrectangle (in zonotope notation) with center 0: $\mathcal{S} Z = (0, v^{(1)}, \ldots, v^{(n)})$, where*

$$v_j^{(i)} = \begin{cases} 0, & i \neq j \\ \overline{S}_j (|c| + \sum_{k=1}^{e} |g^{(k)}|), & i = j, \end{cases}$$

and the subscript j of $v_j^{(i)}$ denotes the j^{th} element of $v^{(i)}$ and S_j denotes the j^{th} row of S.

PROOF. The multiplication of a symmetric interval matrix $\mathcal{S}$ with a zonotope Z is overapproximated by $\mathcal{S}\,Z \subseteq \mathcal{S}\,\mathtt{box}(Z)$ and $\mathtt{box}$ returns an enclosing axis-aligned box which is computed as proposed in [9] as

$$\mathtt{box}(Z) = [c - \Delta g, c + \Delta g], \quad \Delta g = \sum_{i=1}^{e} |g^{(i)}|.$$

It remains to compute $\mathcal{S}[c - \Delta g, c + \Delta g]$, which returns a symmetric interval vector due to the symmetry of $\mathcal{S}$. The upper bound is obtained by $\mathcal{S}\max(|c-\Delta g|, |c+\Delta g|)$. Since $\max(|c - \Delta g|, |c + \Delta g|) = |c| + |\Delta g|$, it follows that

$$\mathcal{S}\,\mathtt{box}(Z) = [-S(|c| + |\Delta g|), S(|c| + |\Delta g|)].$$

Rewriting this result in zonotope notation with generators $v^{(i)}$ yields the result of the proposition. □

We now consider the application of a set of linear transformations, represented as a matrix zonotope, to a set of vectors represented as a zonotope.

PROPOSITION 4 (MATRIX ZONOTOPE MULTIPLICATION). *The product of a matrix zonotope $\mathcal{L}^{[z]} = \{L^{(0)} + \sum_{i=1}^{\kappa} p_i\, L^{(i)} | \; p_i \in [-1, 1]\}$ and a zonotope $Z = (c, g^{(1)}, \ldots, g^{(e)})$ is overapproximated by*

$$\begin{aligned}\mathcal{L}^{[z]}\, Z &= \bigcup_{p_i \in [-1,1]} \left(L^{(0)}\, Z \oplus \bigoplus_{i=1}^{\kappa} p_i\, L^{(i)}\, Z \right) \\ &\subseteq (L^{(0)}c, L^{(0)}g^{(1)}, \ldots, L^{(0)}g^{(e)}, \\ &\quad L^{(1)}c, L^{(1)}g^{(1)}, \ldots, L^{(1)}g^{(e)}, \ldots, \\ &\quad L^{(\kappa)}c, L^{(\kappa)}g^{(1)}, \ldots, L^{(\kappa)}g^{(e)}).\end{aligned}$$

PROOF. The result follows directly from the addition and multiplication rule of zonotopes; see (8). □

The multiplication of a matrix polytope with matrix vertices $V^{(1)}, \ldots, V^{(r)}$ and a zonotopes can be performed as $\mathcal{L}^{[p]}\, Z = \mathtt{CH}(V^{(1)}Z, V^{(2)}Z, \ldots, V^{(r)}Z)$. The result is no longer a zonotope in general, so that it has to be overapproximated by a zonotope. The overapproximation of polytopes by zonotopes is computationally expensive (see, e.g., [3]), so that matrix polytopes should be overapproximated by matrix zonotopes beforehand, making it possible to apply Prop. 4.

In order to quickly estimate the size of the error in the state transition matrix overapproximation, it is often helpful to compute with norms instead of applying the previously introduced computational techniques. In order to obtain a tight norm bound, the matrix set $\mathcal{A}$ is overapproximated by an interval matrix $\mathcal{A}^{[i]}$ which is split into a nominal and a symmetric[3] part: $\mathcal{A}^{[i]} = A^{[n]} + [-S, S]$. The norm of the distance of the set of state transition matrices to the exponential matrix of the nominal matrix is computed for $\||A^{[n]}| + S\| < \frac{2}{t}$ as

$$\begin{aligned}&\|\overline{\mathcal{M}}(t) - e^{A^{[n]}t}\| \\ &\leq \frac{\|A^{[n]}\|\,\|S\|\frac{t^2}{2}}{\|A^{[n]}\| \cdot \||A^{[n]}| + S\|\frac{t^2}{4} - (\|A^{[n]}\| + \||A^{[n]}| + S\|)\frac{t}{2} + 1} \\ &+ \frac{\|S\|t}{1 - \||A^{[n]}| + S\|\frac{t}{2}}.\end{aligned}$$

The proof is neglected due to space limitations.

[3]Symmetric refers here to the set and not to the matrices it contains.

7. NUMERICAL EXAMPLES

In this section we illustrate the proposed methods for computing reachable sets for uncertain time-varying linear systems. The first example demonstrates the difference in accuracy when computing with matrix zonotopes or interval matrices. The scalability of the approach is also demonstrated. The second example shows the usefulness of the approach for computing reachable sets for a hybrid system with nonlinear continuous dynamics.

7.1 Five Dimensional Example

We consider a standard example from the literature [9,22]. In [9], there are no uncertain system matrices, in [22] the system matrices are bounded by an interval matrix. Here, the system matrices are bounded by a matrix zonotope, whose enclosing interval matrix is exactly the one in [22]: $\dot{x} = A(t)x + u(t)$, $A(t) \in \mathcal{A}^{[z]} = (G^{(0)}, G^{(1)})$, $u(t) \in \mathcal{U} = [-\overline{u}, \overline{u}]$, where $\overline{u} = \begin{bmatrix}0.1 & 0.1 & 0.1 & 0.1 & 0.1\end{bmatrix}^T$ and

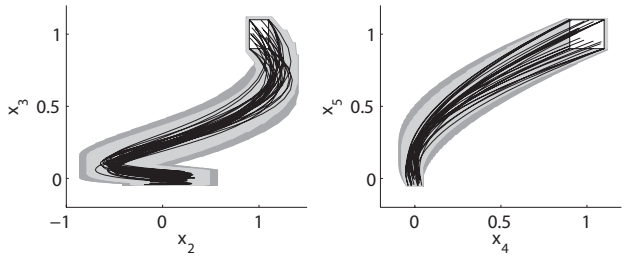
$$G^{(0)} = \begin{bmatrix}-1 & -4 & 0 & 0 & 0 \\ 4 & -1 & 0 & 0 & 0 \\ 0 & 0 & -3 & 1 & 0 \\ 0 & 0 & -1 & -3 & 0 \\ 0 & 0 & 0 & 0 & -2\end{bmatrix}, G^{(1)} = \begin{bmatrix}0.1 & 0.1 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0.1 & 0 \\ 0 & 0 & 0.1 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.1\end{bmatrix}.$$

The reachable set is computed for a step size of $r = 0.05$ and a time horizon of $t_f = 5$. The number of transition matrix terms is chosen as $\eta = 4$ (the first two using matrix zonotopes, the other 2 using interval matrices), and the order of the zonotopes is limited to $\rho = 20$ using the reduction technique in [9]. The computation time is 0.16 s in MATLAB on an i7 Processor and 6GB memory. Plots of selected projections are shown in Fig. 3. It can be seen that the computation is tighter when using matrix zonotopes instead of a tightly enclosing interval matrix. It has not been considered to separate the uncoupled states of this specific example due to the efficiency of the algorithm.

The scalability of the algorithm is shown by computing reachable sets for several linear systems generated with random parameters. Computation times for system matrices bounded by interval matrices and matrix zonotopes are shown in Table 1. Even if the examples contain uncoupled subsystems, the system is computed as if all states are coupled.

7.2 Rollover Verification of a Truck

We consider the problem of determining if a truck will roll over during a set of maneuvers, where the truck has a yaw controller to improve cornering performance. The truck dynamics is velocity dependent, so the yaw controller is a gain scheduling controller that switches among several controllers depending on the velocity. The switching between different controllers is instantaneous rather than cross-fading [20]. Thus, the overall system becomes a hybrid system which is modeled as a hybrid automaton in [3].



(a) Projection onto x_1, x_2. (b) Projection onto x_3, x_4.

Figure 3: Reachable set of the five-dimensional example. The dark gray region shows the reachable set when computing with the interval matrix $\mathcal{A}^{[i]}$, while the light gray region shows the result when computing with the original matrix zonotope $\mathcal{A}^{[z]}$. Black lines show exemplary trajectories and the white region is the initial set.

Table 1: Computation times.

Dimension n	5	10	20	50	100
Interval matrix					
CPU-time [s]	0.10	0.12	0.33	0.82	3.64
Matrix zonotope: Nr of generator matrices $\kappa = 1$					
CPU-time [s]	0.13	0.17	0.60	2.65	8.72
Matrix zonotope: Nr of generator matrices $\kappa = 2$					
CPU-time [s]	0.18	0.30	1.13	4.73	18.77
Matrix zonotope: Nr of generator matrices $\kappa = 4$					
CPU-time [s]	0.34	0.68	2.60	18.07	98.70

We consider two approaches to compute the reachable set. First, the standard approach to hybrid system reachability is applied, where the reachable set computation is continued across discrete transitions using intersections with guard sets. These intersections can introduce significant overapproximation errors; see [3, 11]. Second, as an alternative to the standard approach, the reachable set is computed under a larger set of parameter uncertainties when intersecting several invariant sets. The enlarged set of parameter uncertainties is the union of uncertainties within the invariants the reachable set is intersecting. This makes it possible to compute the reachable set without any intersection operation. This approach is referred to as *continuization* and is beneficial when the intersection operation is dominant in the enlargement of the reachable set, while the effect on computing with a larger set of parameter uncertainties is small as in gain scheduling. This approach is applicable only if the hybrid automaton has no jumps.

The truck dynamics is described by the following continuous state variables (see Fig. 4): the side-slip angle at center of mass β, yaw rate $\dot{\Psi}$, sprung mass roll angle Φ, sprung mass roll angle rate $\dot{\Phi}$, unsprung mass roll angle of the front axle $\Phi_{t,f}$ and the rear axle $\Phi_{t,r}$, and velocity v. The input to the system is the steering angle δ and the longitudinal acceleration a_x. The dynamic equations from [8] are

$$\begin{aligned}
mv(\dot{\beta}+\dot{\Psi}) - m_S h\ddot{\Phi} &= Y_\beta \beta + Y_{\dot{\Psi}}(v)\dot{\Psi} + Y_\delta \delta \\
-I_{xz}\ddot{\Phi} + I_{zz}\ddot{\Psi} &= N_\beta \beta + N_{\dot{\Psi}}(v)\dot{\Psi} + N_\delta \delta \\
(I_{xx}+m_S h^2)\ddot{\Phi} - I_{xz}\ddot{\Psi} &= m_S g h \Phi + m_S v h(\dot{\beta}+\dot{\Psi}) - k_f(\Phi - \Phi_{t,f}) \\
&\quad - b_f(\dot{\Phi} - \dot{\Phi}_{t,f}) - k_r(\Phi - \Phi_{t,r}) - b_r(\dot{\Phi} - \dot{\Phi}_{t,r}) \\
-r(Y_{\beta,f}\beta + Y_{\dot{\Psi},f}\dot{\Psi} + Y_\delta \delta) &= m_{u,f} v(r - h_{u,f})(\dot{\beta}+\dot{\Psi}) + m_{u,f} g h_{u,f} \Phi_{t,f} \\
&\quad - k_{t,f}\Phi_{t,f} + k_f(\Phi - \Phi_{t,f}) + b_f(\dot{\Phi} - \dot{\Phi}_{t,f}) \\
-r(Y_{\beta,r}\beta + Y_{\dot{\Psi},r}\dot{\Psi}) &= m_{u,r} v(r - h_{u,r})(\dot{\beta}+\dot{\Psi}) - m_{u,r} g h_{u,r} \Phi_{t,r} \\
&\quad - k_{t,r}\Phi_{t,r} + k_r(\Phi - \Phi_{t,r}) + b_r(\dot{\Phi} - \dot{\Phi}_{t,r}) \\
\dot{v} &= a_x .
\end{aligned}$$

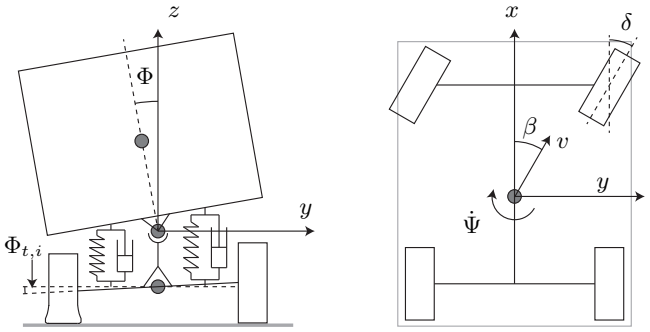


Figure 4: Truck model.

The parameters are chosen as in [8]. In order to obtain a controlled system without steady state error, a PI controller is designed to control the yaw rate $\dot{\Psi}$. Different controllers are active in the intervals $[10, 20 + \Delta v]$, $[20, 30 + \Delta v]$, and $[30, \infty[$ m/s, where $\Delta v > 0$ models the velocity measurement uncertainty. Below 10 m/s, no controller is active. Note that due to the velocity measurement uncertainty, the dynamics is switched in the intervals $[10, 10 + \Delta v]$, $[20, 20 + \Delta v]$, $[30, 30 + \Delta v]$.

The control error is denoted by $e = \dot{\Psi}_d - \dot{\Psi}$, where $\dot{\Psi}_d$ is the desired yaw rate. The desired yaw rate is computed by the steady state solution of $\dot{\Psi}$ when a desired steering angle δ_d or desired lateral acceleration $a_{y,d}$ is used, where $\dot{\Psi}_d = a_{y,d}/v$ in the latter case. The PI controller is written as $\delta = k_1 e + k_2 \int e(t)\, dt$, where k_1 and k_2 are the gains for the proportional and integral part, respectively, which are listed in Table 2.

Table 2: Yaw controller gains.

$v \in$	$[10, 20]$ m/s	$[20, 30]$ m/s	$[30, \infty[$ m/s
controller gains	$k_1 = 0.4$, $k_2 = 1.5$	$k_1 = 0.5$, $k_2 = 2$	$k_1 = 0.6$, $k_2 = 2.5$

The set of possible system matrices is modeled as a matrix zonotope $\mathcal{A}^{[z]}$ and the set of inputs $\mathcal{U}$ as a zonotope. In order to obtain a tight overapproximation of the reachable set, the sets of possible matrices $\mathcal{A}^{[z]}([t_k, t_{k+1}])$ and inputs $\mathcal{U}([t_k, t_{k+1}])$ are updated for each time interval $[t_k, t_{k+1}]$. After introducing the state vector $x = [\beta, \dot{\Psi}, \Phi, \dot{\Phi}, \Phi_{t,f}, \Phi_{t,r}, v, \int e(t)\, dt]^T$ and grouping the terms of the controlled truck

dynamics, one can write them in the form

$$\begin{aligned}\dot{x} =&(p_1Q^{(1)} + p_2Q^{(2)} + p_3Q^{(3)} + p_4Q^{(4)})x \\ &+(p_1R^{(1)} + p_2R^{(2)})a_{y,d},\end{aligned} \tag{11}$$

where for $t \in [t_k, t_{k+1}]$

$$p_1 \in \left[\frac{1}{(\overline{v})^2}, \frac{1}{(\underline{v})^2}\right], \quad p_2 \in \left[\frac{1}{\overline{v}}, \frac{1}{\underline{v}}\right], \quad p_3 = 1 \quad p_4 \in [\underline{v}, \overline{v}],$$

and $\underline{v}, \overline{v}$ are the lower and upper bound of the velocity for $t \in [t_k, t_{k+1}]$. The formulation in (11) gets rid of the nonlinearities and makes it possible to obtain the generators $G^{(i)}$ of the matrix zonotope $\mathcal{A}^{[z]}$ as

$$G^{(0)} = \sum_{i=1}^{4} \texttt{center}(p_i)Q^{(i)}, \text{ for } i = 1..4 : G^{(i)} = \texttt{rad}(p_i)Q^{(i)}$$

and analogously for $\mathcal{B}^{[z]}$, where the operators `center`() and `rad`() return the center and radius of an interval. The set of inputs is obtained as

$$\mathcal{U}([t_k, t_{k+1}]) = \mathcal{B}^{[z]}([t_k, t_{k+1}]) \left[[\underline{a}_{y,d}, \overline{a}_{y,d}] \quad [\underline{a}_x, \overline{a}_x]\right]^T.$$

In order to compute the reachable set under the changing parameter intervals, the computation for $\mathcal{H}_0, \mathcal{P}_0$ in Alg. 1 have to be repeated for each time interval instead of only once as presented in Alg. 1.

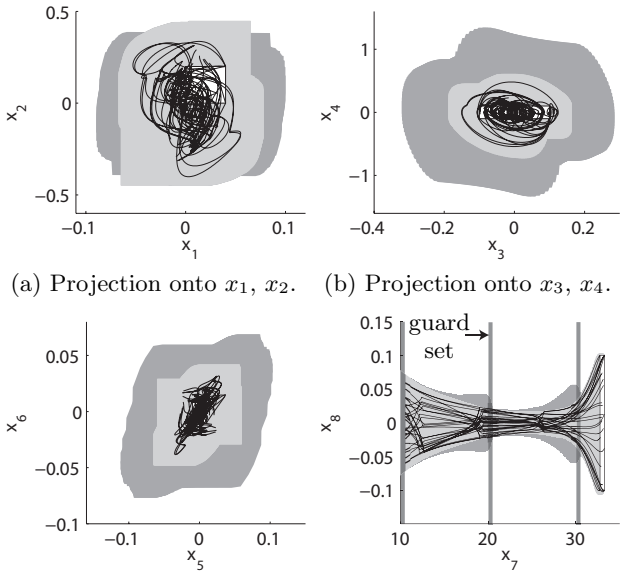
The reachable set is computed for a deceleration maneuver with $a_x = 0.7g$, where g is the gravity constant. Due to limited tire friction, the truck may still perform steering maneuvers that are uncertain within the corresponding set of lateral accelerations $[-a_{y,d}, a_{y,d}]$ and $a_{y,d} = 0.4g$. The reachable set is computed until it has left the half-space of velocities above 10 m/s. Below this velocity, no controller is active anymore. Parameters of the reachable set computation are specified in Table 3. The set of initial states is $x(0) \in [0, 0.04] \times [0, 0.2] \times [-0.1, 0.1] \times [-0.1, 0.1] \times [-0.01, 0.01] \times [-0.01, 0.01] \times [32.75, 33.25] \times [-0.1, 0.1]$.

Table 3: Parameters of the reachable set computation of the truck.

time step size	$r = 0.01$
maximum zonotope order	$\rho = 60$
Taylor series order	$\eta = 4$
velocity measurement uncertainty	$\Delta v = [0, 0.5]$

The reachable set approximations for the deceleration maneuver using the continuization and the hybrid approach are shown in Fig. 5. It can be observed that an enlargement takes place for x_8 after passing the 30, 20, and 10 m/s borders when using the hybrid approach. The reachable set of x_1 and x_2 is almost the same, while the one projected onto $x_3 - x_6$ is much tighter for the continuization approach. The computational time for the hybrid approach is 85 s and 38 s for the continuization approach. The computations have been performed on an Intel i7 Processor with 6GB memory in MATLAB. The truck starts to rollover when the dynamic forces on the rear inner wheel (which is the critical wheel) overcompensate the force due to gravity, which is the case for $x_6 > 0.55$. Thus, the continuization approach verifies the safety of the maneuvers, while the classical approach fails.

The intersection with guard sets has been computed as presented in [3], where zonotopes are transformed to a halfspace representation in order to perform the intersections which are later enclosed by a zonotope to continue the computation with zonotopes. A good compromise between accuracy and efficiency that works surprisingly well for the truck example is to overapproximate the zonotopes by boxes to accelerate the halfspace conversion. The conversion back to zonotopes is done by enclosing the intersected set with two zonotopes in order to increase the accuracy (see [3]). One zonotope is obtained by a principal component technique presented in [26] and the other one by an axis-aligned box.



(a) Projection onto x_1, x_2. (b) Projection onto x_3, x_4.

(c) Projection onto x_5, x_6. (d) Projection onto x_7, x_8.

Figure 5: Reachable set of the yaw controlled truck. The light gray region shows the reachable set for the continuization approach. The dark gray region shows the reachable set for the classical hybrid approach. Black lines show exemplary trajectories of the system.

8. CONCLUSIONS

Previous methods for the reachability analysis of uncertain linear time-invariant systems have been extended to uncertain linear time-varying systems. The presented approach can cope with uncertain system matrices, as well as with arbitrary input trajectories whose values are bounded. In addition, the proposed algorithms scale well with the number of continuous state variables. It has also been demonstrated that the approach can be used for reachability analysis of nonlinear and hybrid systems. The continuization approach is promising for hybrid systems with similar continuous dynamics in adjacent locations. Extensions of the approach to better handle nonlinear dynamics are currently being investigated.

Acknowledgments

This research was supported in part by U.S. National Science Foundation grant number CCF-0926181 and the U.S. Air

Force Office of Scientific Research grant number FA9550-06-1-0312.

9. REFERENCES

[1] M. Althoff. *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars*. PhD thesis, TU München, 2010.

[2] 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*, pages 4042–4048, 2008.

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

[4] E. Asarin, T. Dang, and O. Maler. d/dt: A verification tool for hybrid systems. In *Proc. of the Conference on Decision and Control*, pages 2893–2898, 2001.

[5] A. Chutinan and B. H. Krogh. Computational techniques for hybrid system verification. In *IEEE Transactions on Automatic Control*, volume 48, pages 64–75, 2003.

[6] T. Dang. *Vérification et synthèse des systèmes hybrides*. PhD thesis, Institut National Polytechnique de Grenoble, 2000.

[7] Goran Frehse. PHAVer: Algorithmic verification of hybrid systems past HyTech. In *Hybrid Systems: Computation and Control*, LNCS 3413, pages 258–273. Springer, 2005.

[8] P. Gaspar, I. Szaszi, and J. Bokor. The design of a combined control structure to prevent the rollover of heavy vehicles. *European Journal of Control*, 10:1–15, 2004.

[9] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, LNCS 3414, pages 291–305. Springer, 2005.

[10] A. Girard and C. Le Guernic. Efficient reachability analysis for linear systems using support functions. In *Proc. of the 17th IFAC World Congress*, pages 8966–8971, 2008.

[11] A. Girard and C. Le Guernic. Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Proc. of Hybrid Systems: Computation and Control*, LNCS 4981, pages 215–228. Springer, 2008.

[12] A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In *Hybrid Systems: Computation and Control*, LNCS 3927, pages 257–271. Springer, 2006.

[13] T. Henzinger. *The theory of hybrid automata*, volume 170 of *NATO ASI Series F: Computer and Systems Sciences*, pages 265–292. Springer, 2000.

[14] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE Transactions on Automatic Control*, 43:540–554, 1998.

[15] T. A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi. Beyond HyTech: Hybrid systems analysis using interval numerical methods. In *Hybrid Systems: Computation and Control*, LNCS 1790, pages 130–144. Springer, 2000.

[16] L. Jaulin, M. Kieffer, and O. Didrit. *Applied Interval Analysis*. Springer, 2006.

[17] W. Kühn. Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing*, 61:47–67, 1998.

[18] A. B. Kurzhanskiy and P. Varaiya. Ellipsoidal techniques for reachability analysis of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 52:26–38, 2007.

[19] G. Lafferriere, G. J. Pappas, and S. Yovine. Symbolic reachability computation for families of linear vector fields. *Symbolic Computation*, 32:231–253, 2001.

[20] D. J. Leith and W. E. Leithead. Survey of gain-scheduling analysis and design. *International Journal of Control*, 73:1001–1025, 2000.

[21] N. S. Nedialkov and K. R. Jackson. *Perspectives on Enclosure Methods*, chapter A New Perspective on the Wrapping Effect in Interval Methods for Initial Value Problems for Ordinary Differential Equations, pages 219–264. Springer-Verlag, 2001.

[22] N. Ramdani, N. Meslem, and Y. Candau. Reachability analysis of uncertain nonlinear systems using guaranteed set integration. In *Proc. of the 17th IFAC World Congress*, pages 8972–8977, 2008.

[23] N. Ramdani, N. Meslem, and Y. Candau. Reachability of uncertain nonlinear systems using a nonlinear hybridization. In *Hybrid Systems: Computation and Control*, LNCS 4981, pages 415–428. Springer, 2008.

[24] N. Ramdani, N. Meslem, and Y. Candau. Computing reachable sets for uncertain nonlinear monotone systems. *Nonlinear Analysis : Hybrid Systems*, 4:263–278, 2010.

[25] W. J. Rugh. *Linear System Theory*. Prentice Hall, 1996.

[26] O. Stursberg and B. H. Krogh. Efficient representation and computation of reachable sets for hybrid systems. In *Hybrid Systems: Computation and Control*, LNCS 2623, pages 482–497. Springer, 2003.

[27] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi. Computational techniques for the verification and control of hybrid systems. In *Proceedings of the IEEE*, volume 91, pages 986–1001, 2003.