# Verified Polynomial Controller Synthesis for Disturbed Nonlinear Systems

**Victor Gaßmann and Matthias Althoff**

*Technische Universität München, Boltzmannstr. 3, 85748 Garching (e-mails: {victor.gassmann,althoff}@in.tum.de).*

**Abstract:** We introduce a new control synthesis approach to solve reach-avoid problems occurring in many fields, such as autonomous driving or robot motion planning. Our control approach steers a set of initial states as close as possible to a given target state while provably satisfying constraints on both inputs and states. We compute a control law consisting of a state-dependent, piecewise constant feedforward controller and a continuous feedback controller. The feedforward controller can be polynomial in the state and steers the initial set of states as close as possible to the target state for an undisturbed system, while the feedback controller minimizes the effect of disturbances and abstraction errors. Compared to other formal synthesis approaches, our approach can be verified in polynomial time and generates nonlinear feedforward control laws. The achievable control performance is demonstrated by two use cases.

*Keywords:* Reachability analysis, control synthesis, reach-avoid problems, and optimization.

## 1. INTRODUCTION

Emerging systems – such as autonomous systems – are increasingly complex and safety-critical, making it increasingly difficult to design them only with expert knowledge. In this work, we propose a formal synthesis approach for nonlinear systems to solve reach-avoid problems so that human error in the controller design can be excluded. An example of such a problem is the task of steering an autonomous vehicle to a given target state while avoiding other traffic participants and staying within acceleration limits. To solve this task during online operation, we focus on synthesizing controllers for motion primitives offline, which reduces the online task to suitably concatenating these motion primitives (see e.g. Schürmann and Althoff (2021)).

*Related Work* One popular approach to solving reach-avoid problems is model predictive control (MPC), which solves constrained optimization problems in a moving-horizon control scheme (Alessio and Bemporad, 2009). Since the original MPC problem cannot provably ensure constraints in the presence of disturbances, tube-based MPC keeps the disturbed system within a tube around a reference trajectory (Limon et al., 2010; Mayne et al., 2011). Instead of computing a (possibly) nonlinear optimization problem online, explicit MPC solves the optimization problem up-front (de la Pena et al., 2004; Liu et al., 2012), allowing for much faster computation times compared to implicit MPC; however, the computational effort of getting such an explicit solution with respect to the number of continuous state variables often becomes exponential (Alessio and Bemporad, 2009), and further does not allow one to integrate uncertain initial states.

Another approach to solving reach-avoid problems makes use of linear quadratic regulator (LQR) trees, which are computed using LQR controllers along candidate trajectories, where the region of attraction is conservatively estimated (Tedrake, 2009; Reist and Tedrake, 2010). However, this approach is based on sum-of-squares programming, which – while being a convex programming problem – does not scale well with the system dimension. Controller synthesis using abstraction-based methods (Kloetzer and Belta, 2008; Girard and Martin, 2008; Girard, 2012) is another approach that is able to deal with nonlinear systems as well as input and state constraints. Popular amongst those is symbolic model abstraction, where a smooth continuous system is abstracted to a symbolic model, which greatly simplifies the controller synthesis (Tabuada, 2009; Zamani et al., 2011; Pola and Tabuada, 2009; Pola et al., 2008). However, all these methods require partitioning of the state space, making them intractable for higher-dimensional systems.

Recently, the verification of systems using reachability analysis, which computes a set of states reachable from a given initial set (Althoff, 2010), has gained attention. In Schürmann and Althoff (2017b), the authors synthesize a continuous tracking controller that steers all initial states of a disturbed linear system as close as possible to a target state by incorporating reachability analysis into an optimization problem. In Schürmann and Althoff (2017a), the authors compute parameterized reachable sets to synthesize controllers by solving only a single linear program. While this is computationally efficient, the controller performance is impaired when disturbances dominate. Thus, the approaches from Schürmann and Althoff (2017a,b) are combined in Schürmann and Althoff (2021) to compute a control law with a state-dependent feedforward controller

⋆ We gratefully acknowledge financial support by the projects justITSELF and interACT funded by the European Research Council (ERC) under grant 817629 and 723395, respectively.
⋆⋆Digital Object Identifier: 10.1016/j.ifacol.2021.08.479

as well as continuous feedback controller. Another approach that uses reachability analysis to synthesize a controller combining feedforward and feedback control is described in Verdier et al. (2020), where a controller meeting signal temporal logic specifications is synthesized. While the computed feedforward control is not able to provide a state-dependent feedforward control as in Schürmann and Althoff (2021), it has the benefit that the number of controllers is independent of the sampling time.

*Contributions* The approach described in Schürmann and Althoff (2021) enables robust control synthesis for nonlinear systems. However, since only a linear feedforward controller based on a linear abstraction of the system is computed, it might be far from optimal if the set of initial states is large or the system has a larger degree of nonlinearity. Therefore, we present a novel approach that, to the best of our knowledge, synthesizes for the first time a polynomial state-dependent feedforward control law based on higher-order system abstractions that is combined with a continuous feedback law analogous to Schürmann and Althoff (2021) for solving reach-avoid problems.

*Organization* First, we introduce necessary notation and preliminaries in Sec. 2, followed by an introduction of the problem in Sec. 3. In Sec. 4, we introduce our new approach for computing the polynomial feedforward control. A comparison to the approach from Schürmann and Althoff (2021) as well as a complexity analysis is done in Sec. 5. Lastly, in Sec. 6 we evaluate our new approach.

## 2. NOTATION AND PRELIMINARIES

We introduce required notation and define the concepts of (polynomial) zonotopes, which are subsequently used for reachability analysis as well as controller synthesis.

### 2.1 Notation

For two vectors $w, v \in \mathbb{R}^o$, we define $w^{v^T} = w^v = \prod_{i=1}^{o} w_i^{v_i}$. Given two sets $\mathcal{X}_1$, $\mathcal{X}_2$, their Minkowski addition is denoted by $\mathcal{X}_1 \oplus \mathcal{X}_2$. A zonotope over-approximation of a given set $\mathcal{M}$ is denoted by $Z(\mathcal{M})$. We denote the absolute value of each component of a given vector $x \in \mathbb{R}^p$ by $|x| \in \mathbb{R}^p$. The set of natural numbers $\mathbb{N}$ in this paper includes zero, while $\mathbb{N}_+$ denotes the set of strictly positive, natural numbers. For a given matrix $M$, we denote with $[M]_{(:)}$ the vector that results from collapsing $M$ column-first, i.e., vertically concatenating its column vectors. For a vector $v \in \mathbb{R}^o$, we use $\operatorname{diag}(v)$ to construct a matrix with $v$ as its diagonal elements.

### 2.2 Definitions

A common set representation for reachability analysis are zonotopes.

*Definition 1.* (Zonotope). A zonotope $\mathcal{Z} = \langle c, G \rangle$ with generator matrix $G \in \mathbb{R}^{n \times m}$ and center $c \in \mathbb{R}^n$ is given by

$$\mathcal{Z} = \{x \in \mathbb{R}^n \mid x = c + G\nu, \|\nu\|_\infty \leq 1\}. \tag{1}$$

For convenience, the center and generator matrices of a given zonotope $\mathcal{Z}$ are referred to by $c_{\mathcal{Z}}$ and $G_{\mathcal{Z}}$, respectively.

In this paper, we compute reachability analysis using polynomial zonotopes (Althoff, 2013; Kochdumper and Althoff, 2019), which are a generalization of zonotopes. To that end, we first introduce the concept of a generating function.

*Definition 2.* (Set Generation). We define

$$\{r(\nu)\}_\nu = \{r(\nu) \mid \|\nu\|_\infty \leq 1\} = \mathcal{S},$$

where $r(\nu)$ is the generating function of $\mathcal{S}$. We say that $S = \{r(\nu)\}_\nu$ is generated by $r(\nu)$ over $\nu$.

Next, we define polynomial zonotopes (Kochdumper and Althoff, 2019).

*Definition 3.* (Polynomial Zonotope). Let $c \in \mathbb{R}^n$ be the center, $G = \left[g^{(1)} \cdots g^{(m)}\right] \in \mathbb{R}^{n \times m}$ the generator matrix with generators $g^{(i)} \in \mathbb{R}^n$, $i \in \{1, ..., m\}$, and $E = \left[e^{(1)} \cdots e^{(m)}\right] \in \mathbb{R}^{d \times m}$ the exponent matrix of a polynomial zonotope $\mathcal{PZ}$. We define the generating function (see Def. 2) of a polynomial zonotope as

$$PZ(\nu) = c + \sum_{i=1}^{m} g^{(i)} \nu^{e^{(i)}}.$$

The polynomial zonotope is then given by $\mathcal{PZ} = \{PZ(\nu)\}_\nu$.

## 3. PROBLEM STATEMENT

We are given the system $\dot{x} = f(x, u, w)$, where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state, $u \in \mathcal{U} \subseteq \mathbb{R}^q$ is the controllable input, and $w \in \mathcal{W} \subseteq \mathbb{R}^v$ is some unknown disturbance; here, $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{W}$ are bounded by zonotopes. We make no assumptions about the statistical nature of $\mathcal{W}$.

Our goal is to steer the set of initial states as close as possible to the target state while satisfying constraints (see Schürmann and Althoff (2021)):

$$\begin{aligned} u_{\text{ctrl}}(x,t) = \underset{u(x,t) \in \mathcal{U}}{\arg\min} \; & \max_x \|x - x_{\text{f}}\|_1, \\ \text{s.t. } & x \in \mathcal{R}_{\mathcal{X}^{(0)}, u(x,t), \mathcal{W}}(t_{\text{f}}), \\ & \mathcal{R}_{\mathcal{X}^{(0)}, u(x,t), \mathcal{W}}([0; t_{\text{f}}]) \subseteq \mathcal{X}, \end{aligned} \tag{2}$$

where $x_{\text{f}}$ is a given target state, $t_{\text{f}}$ denotes the final time, and $\mathcal{R}_{\mathcal{X}^{(0)}, u(x,t), \mathcal{W}}(\tau)$ is the set of states that are reachable within a given time frame $\tau$ for a control law $u(x,t) \in \mathcal{U}$ and any disturbance $w \in \mathcal{W}$, starting from the initial set $\mathcal{X}^{(0)}$. Throughout this paper, we assume that constraints are chosen such that the feasible set of (2) is not empty.

Analogously to Schürmann and Althoff (2021), we use feedforward and feedback control, i.e.

$$u_{\text{ctrl}}(x,t) = u_{\text{ff}}(x(0), t) + u_{\text{fb}}(x,t), \tag{3}$$

where $u_{\text{ff}}(x(0), t)$ is the state-dependent feedforward control; $u_{\text{fb}}(x,t) = K(t)(x(t) - x_{\text{ff}}(t))$ is the continuous state feedback with gain matrix $K(t) \in \mathbb{R}^{q \times n}$ and reference trajectory $x_{\text{ff}}(t)$. Since the continuous feedback in this paper is analogous to Schürmann and Althoff (2021), we focus on our novel feedforward synthesis approach, where, analogously to Schürmann and Althoff (2021), we use $\mathcal{U}_{\text{ff}} \subseteq \mathcal{U}$ to reserve input capacity for the feedback controller, and $\mathcal{X}_{\text{ff}} \subseteq \mathcal{X}$ to account for bloating not present in the undisturbed system. Due to space limitations and to improve readability, we assume that $\mathcal{U}_{\text{ff}}$ and $\mathcal{X}^{(0)}$ are given as parallelotopes.

## 4. SYNTHESIS USING REACHABLE SETS

In this section, we compute the feedforward control $u_{\text{ff}}(x(0), t)$. We first describe the controller template and then choose its constant offset such that the center of our initial set is steered to the target state. Using this template, we then compute an approximate reachable set parameterized by our control parameters, which are then obtained by minimizing its size.

### *4.1 Controller Template*

Similarly to Schürmann and Althoff (2021), we assume that $u_{\text{ff}}(x(0), t)$ consists of $N \in \mathbb{N}_+$ piecewise constant control laws. Thus, $\forall t \in \tau^{(i)} = [i; i+1]\frac{t_{\text{f}}}{N}$, $i \in \{0, ..., N-1\}$, we have to make sure that $u_{\text{ff}}^{(i)}(x(0)) \in \mathcal{U}_{\text{ff}}$, or equivalently, $\left|\alpha^{(i)}(x(0))\right| \leq 1$, since

$$\mathcal{U}_{\text{ff}} = \{u \in \mathbb{R}^q \mid u = G_{\mathcal{U}_{\text{ff}}}\alpha + c_{\mathcal{U}_{\text{ff}}},\ \|\alpha\|_\infty \leq 1\}.$$

For the remainder of this section, we drop the index $i$ when convenient.

As will be seen subsequently, it is advantageous to parameterize $u_{\text{ff}}(x(0))$ not in $x(0)$ but in the factors $\beta \in [-1; 1]^n$ of $\mathcal{X}^{(0)}$ to ensure that $u \in \mathcal{U}_{\text{ff}}$. After introducing a user-specified matrix of exponents $O = \left[o^{(1)} \cdots o^{(M)}\right] \in \mathbb{N}^{n \times M}$ and the controller parameters $P = \left[p^{(1)} \cdots p^{(M)}\right] \in [-1; 1]^{q \times M}$, we parameterize $\alpha$ as

$$\alpha(\beta, P) = \sum_{k=1}^{M} p^{(k)} \beta^{o^{(k)}}. \tag{4}$$

One possible choice for $O$ are all monomial exponents up to the maximum desired order $\pi \in \mathbb{N}$. Without loss of generality, we assume that $O$ always contains the all-zero exponent vector to model offsets.

In general, computing the exact range of $\alpha(\beta, P)$ for a fixed $P$ with varying $\beta$ is NP-hard (Gaganov, 1985). That said, we are able to find a feasible parameter set that guarantees constraint satisfaction, albeit with some conservatism when $\alpha(\beta, P)$ is not linear in $\beta$.

*Lemma 4.* (Feasible Parameter Set). Denote by $\mathcal{A}(P) = \{\alpha(\beta, P)\}_\beta$ the polynomial zonotope with $\alpha(\beta, P)$ as in (4). Then, with

$$\mathcal{P} = \{P \mid Z(\mathcal{A}(P)) \subseteq [-1; 1]^q\},$$

which is expressible as a set of linear inequalities, constraint satisfaction is guaranteed, i.e.

$$\forall P \in \mathcal{P}\ \mathcal{A}(P) \subseteq [-1; 1]^q.$$

Further, for $\alpha(\beta, P)$ linear in $\beta$, $\mathcal{P}$ parameterizes the available space completely, i.e.

$$\bigcup_{P \in \mathcal{P}} \mathcal{A}(P) = [-1; 1]^q.$$

*Proof.* $\mathcal{P}$ guarantees constraint satisfaction because $\forall P \in \mathcal{P}\ \{\alpha(\beta, P)\}_\beta \subseteq Z(\mathcal{A}(P)) \subseteq [-1; 1]^q$. Further, let $Z(\mathcal{A}(P)) = \left\langle c(P), \left[g^{(1)}(P) \ \dots \ g^{(L)}(P)\right]\right\rangle$. Then it is true that $Z(\mathcal{A}(P)) \subseteq [-1; 1]^q \iff |c(P)| + \sum_{j=1}^{L}\left|g^{(j)}(P)\right| \leq 1$; by introducing auxiliary variables, $\mathcal{P}$ is thus representable by a system of linear inequalities.

Lastly, let $\alpha(\beta, P)$ be linear in $\beta$. Then $Z(\mathcal{A}(P)) = \mathcal{A}(P)$ and hence $\forall a \in [-1; 1]^q\ \exists P \in \mathcal{P}\ \exists \beta \in [-1; 1]^n\ \alpha(\beta, P) = a$, which concludes the proof. □

While the parameterization of the controller template in $\beta$ (see (4)) is advantageous for computing its set of feasible parameters in Lemma 4, we need our controller dependent on $x(0)$ (see (3)) in order to apply it in practice. Since $\mathcal{X}^{(0)} = \langle c_{\mathcal{X}^{(0)}}, G_{\mathcal{X}^{(0)}}\rangle$ is a parallelotope, $G_{\mathcal{X}^{(0)}}^{-1}$ exists and $\beta = G_{\mathcal{X}^{(0)}}^{-1}(x(0) - c_{\mathcal{X}^{(0)}})$, so that we can write

$$u_{\text{ff}}(x(0)) = c_{\mathcal{U}_{\text{ff}}} + G_{\mathcal{U}_{\text{ff}}}\alpha\left(G_{\mathcal{X}^{(0)}}^{-1}(x(0) - c_{\mathcal{X}^{(0)}}), P\right). \tag{5}$$

As described in Schürmann and Althoff (2021), we can compute reference inputs $u_c^{(0:N-1)} = c_{\mathcal{U}_{\text{ff}}} + G_{\mathcal{U}_{\text{ff}}}\alpha_c^{(0:N-1)}$ that steer $c_{\mathcal{X}^{(0)}}$ to $x_{\text{f}}$. By setting $p^{(k)} = \alpha_c^{(i)}$ and $o^{(k)} = 0$, where $k \in \{1, ..., M\}$, $u_{\text{ff}}^{(i)}(c_{\mathcal{X}^{(0)}}) = u_c^{(i)}$ (see (4)) after inserting $c_{\mathcal{X}^{(0)}}$ into (5). Next, we compute the closed-loop reachable set at $t = t_{\text{f}}$ to tune these control parameters.

### *4.2 Parameterized Closed-Loop Reachable Set*

For efficient synthesis, we introduce an approximation of the undisturbed closed-loop reachable set

$$\mathcal{R}^{(N)}\left(P^{(0:N-1)}\right) = \left\{R^{(N)}\left(\beta, P^{(0:N-1)}\right)\right\}_\beta, \tag{6}$$

at time $t = t_{\text{f}}$, where $R^{(N)}\left(\beta, P^{(0:N-1)}\right)$ is the generating function according to Def. 2, for the controller parameterization $P^{(0:N-1)} = \left[P^{(0)} \ \dots \ P^{(N-1)}\right]$ for all $N$ piecewise constant controllers. This approximate reachable set is computed by truncating the Taylor expansion of our undisturbed system dynamics $f(x, u)$ at a desired order $\kappa \in \mathbb{N}_+$ (Althoff, 2013, Sec. 3.1, Eq. 2). The difference between the approximation $\mathcal{R}^{(N)}\left(P^{(0:N-1)}\right)$ and the exact reachable set becomes small (Althoff, 2013, Sec. 4.2) for smaller time steps and a higher order $\kappa$.

### *4.3 Controller Computation*

To arrive at a tractable problem, we adapt the feedforward controller optimization from Schürmann and Althoff (2021) by first computing a zonotope over-approximation of the parameterized reachable set in (6) for $t = t_{\text{f}}$ in $\beta$, i.e., $Z\left(\mathcal{R}^{(N)}(P)\right) = \langle c_{\mathcal{R}^{(N)}}(P), G_{\mathcal{R}^{(N)}}(P)\rangle$. With $P^{(i)} = \left[\alpha_c^{(i)},\ p^{(i,2)},\ \dots\ p^{(i,M)}\right]$ from Sec. 4.1, we choose the optimal matrix of control parameters as

$$\begin{aligned} \hat{P} = \ & \underset{P}{\arg\min} \ \left\|\left[G_{\mathcal{R}^{(N)}}(P)\right]_{(:)}\right\|_1 + \mu\left\|[P]_{(:)}\right\|_1, \\ & \text{s.t. } P = P^{(0:N-1)}, \\ & \qquad P^{(i)} \in \mathcal{P},\ \forall i \in \{0, ..., N-1\}, \\ & \qquad \mathcal{R}^{(N)}(P) \subseteq \mathcal{X}_{\text{ff}}, \end{aligned} \tag{7}$$

where $\mathcal{R}^{(N)}(P) \subseteq \mathcal{X}_{\text{ff}}$ can, for example, be checked using Schürmann and Althoff, 2021, Lemma 2, and $\mu \geq 0$ is a user-specified penalty for large inputs. While $\mathcal{R}^{(N)}(P) \subseteq \mathcal{X}_{\text{ff}}$ checks the tightened state constraints only at a discrete point in time, we verify that state constraints $\mathcal{X}$ are met at all times during feedback control optimization (see (2)).

The feedforward control for each step $i \in \{0, ..., N-1\}$ is then obtained by substituting $\hat{P} = \hat{P}^{(0:N-1)}$ into (5), i.e.

$$u_{\text{ff}}^{(i)}(x(0)) = c_{\mathcal{U}_{\text{ff}}} + G_{\mathcal{U}_{\text{ff}}} \alpha \left( G_{\mathcal{X}^{(0)}}^{-1} \left( x(0) - c_{\mathcal{X}^{(0)}} \right), \hat{P}^{(i)} \right).$$

Subsequently, we discuss the properties of our approach.

## 5. DISCUSSION OF THE ALGORITHM

In this section, we discuss two noteworthy aspects of our approach: First, we show in Sec. 5.1 that our approach is a generalization of Schürmann and Althoff (2021) under the parallelotope assumption of $U_{\text{ff}}$, $\mathcal{X}^{(0)}$ made in Sec. 3. In Sec. 5.2 and Sec. 5.3, we analyze our algorithm with respect to its online and offline complexity, respectively.

### *5.1 Generalization of Combined Control*

We only have to show the generalization of the feedforward controller, since the feedback controller is identical. In the following, we explicitly consider the index $i$ for the control parameters.

We start by proving that for $\kappa = 1$ and a control template linear in $\beta$ with $o^{(1)} = 0$, i.e.

$$u_{\text{ff}}\left(\beta, P^{(0)}\right) = c_{\mathcal{U}_{\text{ff}}} + G_{\mathcal{U}_{\text{ff}}} \left( p^{(0,1)} + \sum_{k=1}^{n} p^{(0,k+1)} \beta_k \right), \quad (8)$$

we arrive at the same cost function as in Schürmann and Althoff, 2021, Theorem 1.

To compute the parameterized reachable set at time $t = t_{\text{f}}$, we formulate a new state $z = \left[x^T \ u^T\right]^T$ with a new initial set $\mathcal{Z}^{(0)}\left(P^{(0)}\right) = \left\{ \begin{bmatrix} c_{\mathcal{X}^{(0)}} + G_{\mathcal{X}^{(0)}}\beta \\ u_{\text{ff}}\left(\beta, P^{(0)}\right) \end{bmatrix} \right\}_\beta$. The corresponding dynamics are $\dot{z} = \hat{f}(z) = \left[f(x,u)^T \ 0^T\right]^T$. Let $\hat{A} = \left.\frac{d\hat{f}(z)}{dz}\right|_{z=\bar{z}} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}$, where $\bar{z} = \left[\bar{x}^T \ \bar{u}^T\right]^T$ is the linearization point of the extended system $\hat{f}(z)$, $A = \left.\frac{\partial f(x,u)}{\partial x}\right|_{\substack{x=\bar{x}\\u=\bar{u}}}$, and $B(\bar{z}) = \left.\frac{\partial f(x,u)}{\partial u}\right|_{\substack{x=\bar{x}\\u=\bar{u}}}$. The reachable set at time $t = r$ with $r = \frac{t_{\text{f}}}{N}$ is then given by

$$\hat{\mathcal{R}}^{(1)}\left(P^{(0)}\right) = e^{\hat{A}r} \mathcal{Z}^{(0)}\left(P^{(0)}\right). \quad (9)$$

By using

$$e^{\hat{A}r} = \begin{bmatrix} \sum_{k=0}^{\infty} \frac{(Ar)^k}{k!}, & \sum_{k=1}^{\infty} \frac{A^{k-1}r^k}{k!} B \\ 0, & I \end{bmatrix} = \begin{bmatrix} e^{Ar}, & \int_0^r e^{A\tau} d\tau B \\ 0, & I \end{bmatrix},$$

(9) becomes

$$\mathcal{R}^{(1)}\left(P^{(0)}\right) = e^{Ar}\mathcal{X}^{(0)} \oplus \int_0^r e^{A\tau} d\tau B \left\{ u_{\text{ff}}\left(\beta, P^{(0)}\right) \right\}_\beta, \quad (10)$$

which is identical to what is given in Schürmann and Althoff, 2021, Lemma 1 (proof) for the first step.

By iterative application of the above, the parameterized reachable set $\mathcal{R}^{(N)}\left(P^{(0:N-1)}\right)$ is thus identical to Schürmann and Althoff (2021). Because $\kappa = 1$, this reachable set is represented by a zonotope and thus (7) is identical to the optimization problem in Schürmann and Althoff, 2021, Theorem 1, resulting in the same controller.

### *5.2 Offline Complexity*

While we cannot provide the overall complexity due to the usage of nonlinear programming, we subsequently discuss the offline complexity of different components of our approach.

*Reachability Analysis* For a fixed $\kappa$, reachability analysis using Althoff (2013) is polynomial in the number of state variables $n$ when appropriate reduction methods are used.

*Center Trajectory* Since the center trajectory is computed analogously to Schürmann and Althoff (2017a), no general complexity bound is available. However, efficient methods exist (Betts, 2010).

*Feedforward Controller Optimization* Since (7) is a nonlinear programming problem, there is no general bound for its complexity. In practice however, it can be efficiently solved to a local optimum, since both the objective function as well as all constraints can be expressed as polynomials, and thus Jacobians and Hessians can be computed in polynomial time.

*Feedforward Control Parameters and Polynomial Composition* The complexity of evaluating the feedforward controller in (5) is dominated by the complexity of polynomial composition. For a fixed controller order, the number of monomials of the result is polynomially upper bounded for each input dimension, and each monomial requires a multiplication of at most $n$ polynomials of fixed degree, raised to a power of at most $\pi$. Thus, the computational complexity is polynomial in the number of state variables.

### *5.3 Online Complexity*

In contrast to the computational complexity for the offline computations, we can show that the online complexity is polynomial in the number of state variables.

*Feedforward Control* Computing the control feedforward input at $t \in \tau^{(i)}$ requires evaluating (5), which is of polynomial complexity in the number of state variables.

*Feedback Control* Denote the abstraction of our undisturbed system in Sec. 4.2 by $\dot{x} = h(x,u)$. The feedforward trajectory is then obtained as the solution to $\dot{x}_{\text{ff}} = h(x_{\text{ff}}, u_{\text{ff}}(x(0), t))$. Thus, online application of the controller requires simulation of $x_{\text{ff}}(t)$, which can be achieved in polynomial time by, for example, using MATLAB's Runge-Kutta-based solver ode45 [1]. Further, we need to execute one matrix-vector multiplication as well as one vector-vector addition in (3), which are also of polynomial complexity.

[1] `https://mathworks.com/help/matlab/ref/ode45.html`

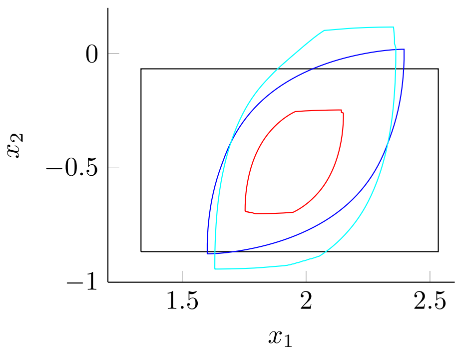


Fig. 1. Reachable sets of our novel approach with $\kappa = 2$, $\pi = 2$ (red) and the feedforward reachable set from Schürmann and Althoff (2021) using zonotopes (blue) and polynomial zonotopes (cyan) for the van-der-Pol system.

## 6. RESULTS

We use the CORA toolbox (Althoff, 2015) for reachability analysis and the AROC toolbox [2] for controller synthesis. All simulations are run on an Intel i7-9700k with 16GB of RAM and MATLAB 2020b with parallel computing. We denote the final, closed-loop reachable set by $\mathcal{R}_{\text{cl}}(\cdot)$, which is attained by applying the synthesized controller. We first synthesize a controller for a variant of the van-der-Pol system in Sec. 6.1, and then synthesize a controller for a high-fidelity single-track model in Sec. 6.2.

### *6.1 Van-der-Pol Oscillator*

A controlled van-der-Pol system can be described by

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = \left(1 - x_1^2\right) x_2 - x_1 + u + w,$$

where $x \in \mathbb{R}^2$ is the state, $u \in \mathbb{R}$ denotes the control input, and $w \in \mathbb{R}$ is the disturbance to the system. The initial set is given by $\mathcal{X}^{(0)} = \left\langle [1.4 \;\, 2.4]^T, \text{diag}\left([0.6 \;\, 0.4]\right)\right\rangle$, the input set by $\mathcal{U} = \mathcal{U}_{\text{ff}} = \langle 0, 2\rangle$, and the disturbance is given by $\mathcal{W} = \langle 0, 0.1\rangle$. The target state is $x_{\text{f}} = [1.9336 \;\, {-0.4665}]^T$ and the final time is $t_{\text{f}} = 1$. In this example, we compare our novel feedforward control law with that from Schürmann and Althoff (2021), where both consist of $N = 4$ piecewise constant control laws.

Synthesizing a quadratic feedforward controller using our new approach with abstraction order $\kappa = 2$ takes around 7 seconds, while computing the linear feedforward controller from Schürmann and Althoff (2021) as implemented in AROC takes around 4 seconds. Fig. 1 depicts the achieved final reachable sets of the new approach (in red) and the feedforward approach from Schürmann and Althoff (2021), for which we computed the reachable set using both zonotopes (blue) and polynomial zonotopes (cyan). This comparison clearly shows the superiority of our novel approach.

[2] `https://tumcps.github.io/AROC/`

### *6.2 High-Fidelity Single-Track Model*

By modeling vehicles using simplified dynamics, e.g. the kinematic single-track vehicle model (Schürmann and Althoff, 2021), important effects, such as understeering or oversteering, are not considered. Therefore, we use the six-dimensional benchmark example described in Althoff et al., 2017, Sec. 3. Due to large nonlinearities in this system, the approach from Schürmann and Althoff, 2021 in AROC is not able to produce a feasible solution.

The state of this system is given by $x = \left[\beta \;\, \Psi \;\, \dot{\Psi} \;\, v \;\, s_x \;\, s_y\right]^T$, where $\beta$ is the slip angle, $\Psi$ and $\dot{\Psi}$ are the yaw angle and yaw angle rate, respectively, $v$ is the longitudinal velocity, and $s_x, s_y$ denote the two spatial coordinates of the vehicle. Further, the input to the system is given by $u = [\delta \;\, \dot{v}]^T$, where $\delta$ is the steering angle of the front wheel and $\dot{v}$ is the longitudinal acceleration.

We set $\mathcal{X}^{(0)} = \langle c_{\mathcal{X}^{(0)}}, G_{\mathcal{X}^{(0)}}\rangle$ with $c_{\mathcal{X}^{(0)}} = [0 \;\, 0 \;\, 0 \;\, 20 \;\, 0 \;\, 0]^T$ and $G_{\mathcal{X}^{(0)}} = \text{diag}\left([0.2 \;\, 0.02 \;\, 0.2 \;\, 0.2 \;\, 0.2 \;\, 0.2]\right)$. The final target state is $x_{\text{f}} = [0 \;\, 0 \;\, 0 \;\, 20 \;\, 20 \;\, 0]^T$, the available input set is $\mathcal{U} = \langle 0, \text{diag}\left([0.4 \;\, 9.81]\right)\rangle$, and the disturbance set is $\mathcal{W} = \langle 0, \text{diag}\left([0.004 \;\, 0.1]\right)\rangle$. Lastly, let $t_{\text{f}} = 1$, $N = 5$, and $\mathcal{X} = \langle x_{\text{f}}, G_{\mathcal{X}^{(0)}}\rangle$.

Choosing a linear controller based on a second-order abstraction, we compute a feasible solution to (2) in around 4.5 minutes (12.5 seconds for feedforward computation). Fig. 2 shows projections of the closed-loop reachable set that is obtained by applying the synthesized controller to our system. Thanks to the constrained set of states $\mathcal{X}$, the final reachable set is contained in the shifted initial set.

## 7. CONCLUSION

We introduced a new approach to solving reach-avoid problems while provably satisfying input and state constraints. To the best of our knowledge, we synthesize polynomial control laws using reachability analysis for the first time. By not limiting the order of abstraction, the parameterized reachable set more accurately reflects the exact reachable set. This leads to a better control performance, which was validated by our experimental results, albeit at the cost of increased computational effort.

Possible future work should focus on improving the performance for larger input sets, and computing tighter feasible parameter sets for higher-degree control templates. Further, reducing computational effort for the feedback synthesis should be a focus of future research, as it currently accounts for the majority of total computation time.

## REFERENCES

Alessio, A. and Bemporad, A. (2009). A survey on explicit model predictive control. In *Nonlinear model predictive control*, 345–369.

Althoff, M. (2010). *Reachability analysis and its application to the safety assessment of autonomous cars*. Ph.D. thesis, Technische Universität München.

Althoff, M. (2013). Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *Proceedings of the 16th international*

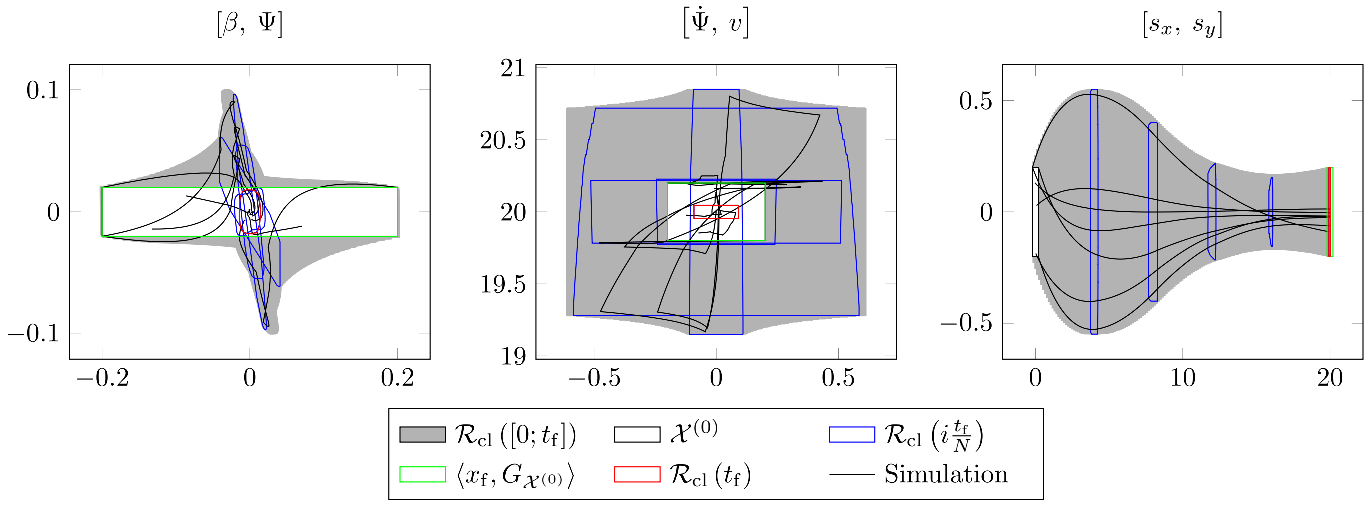


Fig. 2. Projections of the final reachable set for the high-fidelity single-track model from Althoff et al., 2017, Sec. 3, using a linear controller and a second-order abstraction ($\kappa = 2$).


*conference on Hybrid Systems: Computation and Control*, 173–182.

Althoff, M. (2015). An introduction to CORA 2015. *ARCH14-15. 1st and 2nd International Workshop on Applied Verification for Continuous and Hybrid Systems*, 34, 120–151.

Althoff, M., Koschi, M., and Manzinger, S. (2017). CommonRoad: Composable benchmarks for motion planning on roads. In *Proc. of the IEEE Intelligent Vehicles Symposium*, 719–726.

Betts, J.T. (2010). *Practical methods for optimal control and estimation using nonlinear programming*. SIAM.

de la Pena, M., Bemporad, A., and Filippi, C. (2004). Robust explicit MPC based on approximate multiparametric convex programming. In *43rd IEEE Conference on Decision and Control*, 2491–2496.

Gaganov, A. (1985). Computation complexity of the range of a polynomial in several variables. *Cybernetics*, 21(4), 418–421.

Girard, A. (2012). Controller synthesis for safety and reachability via approximate bisimulation. *Automatica*, 48(5), 947–953.

Girard, A. and Martin, S. (2008). Motion planning for nonlinear systems using hybridizations and robust controllers on simplices. In *47th IEEE Conference on Decision and Control*, 239–244.

Kloetzer, M. and Belta, C. (2008). A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1), 287–297.

Kochdumper, N. and Althoff, M. (2019). Sparse polynomial zonotopes: A novel set representation for reachability analysis. *arXiv preprint arXiv:1901.01780*.

Limon, D., Alvarado, I., Alamo, T., and Camacho, E. (2010). Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control*, 20(3), 248–260.

Liu, C., Chen, W.H., and Andrews, J. (2012). Tracking control of small-scale helicopters using explicit nonlinear MPC augmented with disturbance observers. *Control Engineering Practice*, 20(3), 258–268.

Mayne, D.Q., Kerrigan, E.C., Van Wyk, E., and Falugi, P. (2011). Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11), 1341–1353.

Pola, G., Girard, A., and Tabuada, P. (2008). Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10), 2508–2516.

Pola, G. and Tabuada, P. (2009). Symbolic models for nonlinear control systems: Alternating approximate bisimulations. *SIAM Journal on Control and Optimization*, 48(2), 719–733.

Reist, P. and Tedrake, R. (2010). Simulation-based LQR-trees with input and state constraints. In *IEEE International Conference on Robotics and Automation*, 5504–5510.

Schürmann, B. and Althoff, M. (2021). Optimizing sets of solutions for controlling constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 66(3), 981–994.

Schürmann, B. and Althoff, M. (2017a). Guaranteeing constraints of disturbed nonlinear systems using set-based optimal control in generator space. *IFAC-PapersOnLine*, 50(1), 11515–11522.

Schürmann, B. and Althoff, M. (2017b). Optimal control of sets of solutions to formally guarantee constraints of disturbed linear systems. In *2017 American Control Conference*, 2522–2529. IEEE.

Tabuada, P. (2009). *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media.

Tedrake, R. (2009). LQR-trees: Feedback motion planning on sparse randomized trees. In *Proceedings of Robotics: Science and Systems*.

Verdier, C.F., Kochdumper, N., Althoff, M., and Mazo Jr, M. (2020). Formal synthesis of closed-form sampled-data controllers for nonlinear continuous-time systems under STL specifications. *arXiv preprint arXiv:2006.04260*.

Zamani, M., Pola, G., Mazo, M., and Tabuada, P. (2011). Symbolic models for nonlinear control systems without stability assumptions. *IEEE Transactions on Automatic Control*, 57(7), 1804–1809.