Received 15 March 2023; revised 9 June 2023; accepted 17 July 2023. Date of publication 2 August 2023;
date of current version 15 September 2023. Recommended by Senior Editor Majid Zamani.

Digital Object Identifier 10.1109/OJCSYS.2023.3301335

Polynomial Controller Synthesis of Nonlinear Systems With Continuous State Feedback Using Trust Regions

VICTOR GAßMANN AND MATTHIAS ALTHOFF

(Formal Verification and Synthesis of Cyber-Physical Systems)

School of Computation, Information and Technology, Technische Universität München, 85748 Garching, Germany

CORRESPONDING AUTHOR: Victor Gaßmann (e-mail: victor.gassmann@tum.de).

This work was supported by the project justITSELF funded by the European Research Council (ERC) under Grant Agreement 817629.

ABSTRACT We present a novel, correct-by-construction control approach for disturbed, nonlinear systems with continuous state feedback under state and input constraints. For the first time, we jointly synthesize a feedforward and feedback controller by solving a single non-convex, continuously differentiable approximation of the original synthesis problem, which we combine with a trust-region approach in an iterative manner to obtain non-conservative results. We ensure the formal correctness of our algorithm through reachability analysis and show that its computational complexity is polynomial in the state dimension for each trust-region iteration. In contrast to previous work, we also avoid the introduction of several algorithm parameters that require expert knowledge to tune, making the proposed synthesis approach easier to use for non-experts while guaranteeing state and input constraint satisfaction. Numerical benchmarks demonstrate the applicability of our novel synthesis approach.

INDEX TERMS Correct-by-construction controller synthesis, optimization, reachability analysis, trust regions.

I. INTRODUCTION

Many newly developed autonomous systems are safety-critical, such as automated vehicles or robots acting in human environments. As these autonomous systems become more capable, their increased complexity makes it virtually impossible to manually design controllers that always ensure their safety.

A wide range of safety-critical tasks for these autonomous systems can be classified as reach-avoid problems: An agent tries to steer a system to a given target while ensuring constraints on both the input and the state of the system. Human-robot collaboration for pick-and-place tasks or package delivery using drones can be interpreted as such reach-avoid problems. While these tasks have been successfully solved, their operational safety often cannot be guaranteed, which is especially problematic for safety-critical applications.

A. RELATED WORK

To solve these challenging problems while guaranteeing safety, many different approaches – surveyed subsequently – exist.

a) Hamilton-Jacobi equations: Hamilton-Jacobi equations provide a flexible way for computing reachable sets of systems with input and state constraints [1]. A reachable set contains all system states that are reachable within a given time frame for a given initial set and input set. Although approaches using the Hamilton-Jacobi equations generally scale exponentially with the number of states, they are rather popular since they naturally compute optimal controllers for differential games [2] – this is rather difficult to realize with other methods that only solve optimal control problems without any opponent. While these approaches synthesize controllers for systems described by ordinary differential equations, the method from [3] can synthesize robust and optimal feedback

controllers for systems given by nonlinear partial differential equations. In summary, all approaches using Hamilton-Jacobi equations suffer from the curse of dimensionality, induced by the inherent complexity required to solve them.

b) Model abstractions: Another popular branch of research for controller synthesis uses model abstractions. Especially popular among these are symbolic model abstractions, which essentially abstract hybrid, i.e., mixed discrete and continuous dynamics, to a purely discrete system [4], [5], [6], [7], [8], [9], [10], [11], [12]. This makes it possible to directly use the synthesis approaches developed by the computer science community for purely discrete systems. While earlier work was often only applicable to certain system classes [9], [13], [14], quantized input or states [15], [16], or required certain stability assumptions [17], [18], more recent approaches apply to a broader range of system classes and even consider complex task specifications formalized by temporal logic [19], [20], [21], [22], [23], [24]. The abstraction to discrete systems leads to an exponential worst-case complexity in the number of state variables and consequently often requires a prohibitively large amount of memory for higher-dimensional systems. Thus, there has recently been an increased focus on abstraction techniques that do not require state space discretization. However, these techniques either use mixed-integer linear programming (MILP), whose worst-case complexity is exponential in the number of integer variables [25], mainly focus on linear systems [26], or use specific system properties to more efficiently compute abstractions [27].

c) Control Lyapunov and control barrier functions: A possible way to circumvent the exponential complexity of abstraction-based approaches is to synthesize controllers based on Lyapunov-like functions. However, this only translates the controller synthesis problem to a synthesis problem of the required Lyapunov-like functions, which is often not much easier to solve. Mainly two techniques have been proposed so far: Control Lyapunov functions (CLFs) [28] and control barrier functions (CBFs) [29]. Recent methods combine both techniques [30], [31], [32], [33], where CBFs encode constraints to ensure safety and CLFs guarantee the stability of the system. Since, by construction, both can be included in the optimization problem computing the optimal controller, they allow one to synthesize stable controllers adhering to both input and state constraints. To derive these barrier functions, however, all mentioned works restrict the class of nonlinear dynamics in most cases to control-affine systems.

In contrast to formulating CBFs and CLFs manually, the works in [34], [35] solve reach-avoid problems for nonlinear systems by computing linear-quadratic regulators (LQRs) along candidate trajectories, where the region of attraction along these trajectories is estimated using automatically computed Lyapunov functions. While these approaches apply to general nonlinear dynamics, they only provide probabilistic guarantees. The work in [36] uses a similar approach as in [34] for piecewise affine (PWA) systems instead of nonlinear systems. However, the authors use mixed-integer programming, which has exponential worst-case complexity in the system dimension.

d) Model predictive control: Model predictive control (MPC) has been used extensively in industry for the past decades [37], [38] due to its ability to easily handle input and state constraints. Since early variants of MPC were not able to consider disturbances, tube-based MPC was introduced, which essentially keeps the system within a tube around a reference trajectory [39], [40], [41], [42], [43]. In contrast to, e.g., abstraction-based methods, which compute their controllers offline and then simply choose an appropriate controller online, implicit MPC computes the control inputs online. To combat potentially long online computation times due to repeated optimization, explicit MPC aims to solve the optimization problem offline directly [44], [45]; however, its computation often becomes exponential in the number of continuous state variables due to the necessity of state space partitioning when explicitly solving more complex optimization problems. That said, recent advances in convex optimization theory enable real-time applications of robust MPC for selected classes of systems, such as linear systems [46], [47].

e) Reachability analysis: Reachability analysis has gained attention as an efficient tool for verification tasks [48]. In [49], the authors combine reachability analysis and nonlinear optimization to directly synthesize a feedback controller for linear time-invariant systems. Since this approach requires the recomputation of the reachable set in each optimization iteration, the work in [50] computes an approximation of the reachable set, which is directly parameterized in the control parameters to be computed. Thus, it is possible to quickly and efficiently synthesize a linear, piecewise constant controller for general, input-constrained nonlinear systems by a single linear program. Due to the nature of piecewise constant controllers, however, performance degrades with decreasing sampling frequency or when disturbances dominate. To remedy this shortcoming, the authors of [51] combine the feedforward control computation from [50] and the feedback optimization from [49], enabling controller synthesis that is provably correct while respecting input and state constraints at the cost of computing the reachable set in each optimization iteration. In [52], the authors propose a generalization of this combined controller synthesis to realize polynomial feedforward controllers.

B. CONTRIBUTIONS

To circumvent the recomputation of reachable sets during each optimization iteration, we propose an algorithm which – for the first time – combines the synthesis of a piecewise constant, polynomial, state-dependent feedforward controller and a continuous state feedback controller. This combined synthesis avoids the introduction of additional algorithm parameters as in previous works, which need to be tuned by experts. To that end, we combine approximations for the undisturbed feedforward reachable set and the disturbance tube, which enlarges the undisturbed feedforward reachable set, to

formulate a single non-convex, continuously differentiable optimization problem for the combined synthesis. To ensure the accuracy of this approximated optimization problem, we restrict its domain using trust regions, i.e., areas of the control parameters for which we trust the accuracy of the aforementioned approximations. The trust regions are iteratively updated, and we employ overapproximative reachability analysis in each iteration to guarantee formal correctness. Because these overapproximative reachable sets are available after each iteration, we only approximate the relative difference to these known, tight sets. As a result, we are robust against approximation errors because they do not accumulate over the iterations. In summary, our approach is fully automated and only requires the user to provide the number of controllers and the polynomial order of the feedforward controllers in the initial state (e.g., linear or quadratic).

II. PRELIMINARIES

We first introduce necessary notation and define the class of systems considered in this article as well as the concept of reachable sets. We then briefly describe all required set representations.

A. NOTATION

We denote with $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_+$, $\mathbb{N}$, $\mathbb{N}_+$, $\mathbb{S}$, and $\mathbb{S}_{++}$ the sets of real numbers, non-negative real numbers, positive real numbers, natural numbers, positive natural numbers, symmetric matrices, and the cone of positive-definite matrices. The identity matrix of dimension n is denoted by I_n, and we define 1_n and $1_{n\times m}$ as the n-dimensional all-ones vector and the n-by-m-dimensional all-ones matrix, respectively. For two vectors $w \in \mathbb{R}^n$ and $v \in \mathbb{R}^n$, we introduce the multi-index notation $w^v = \prod_{i=1}^n w_i^{v_i}$. We use $\operatorname{diag}(v) \in \mathbb{R}^{n\times n}$ to construct a matrix with $v \in \mathbb{R}^n$ as its diagonal elements. Given a matrix $M \in \mathbb{R}^{m_1\times m_2}$, $M_{(:)} \in \mathbb{R}^{m_1 m_2}$ is the vector that results from stacking all columns of M. Further, for a function $x : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$ and time $t \in \mathbb{R}_{\geq 0}$, we define $\dot{x}(t) = \frac{dx(t)}{dt}$. We denote sets by upper-case calligraphic letters. The Minkowski sum of two sets $\mathcal{A} \subseteq \mathbb{R}^n$ and $\mathcal{B} \subseteq \mathbb{R}^n$ is defined as $\mathcal{A} \oplus \mathcal{B} = \{a + b \mid a \in \mathcal{A},\ b \in \mathcal{B}\}$. Finally, we use $O(\cdot)$ for the Landau notation of the asymptotic computational complexity.

B. DEFINITIONS

In this article, we synthesize controllers of a given system for a set of initial states instead of a single initial state. Hence, we first introduce the considered system class and then use it to introduce the notion of reachable sets.

Definition 1 (System): We consider a system with dynamics $\dot{x}(t) = f(x(t), u(t), w(t)) \in \mathbb{R}^{n_x}$, where f is a twice continuously differentiable function, $x(t) \in \mathbb{R}^{n_x}$ denotes the state of the system at time $t \in \mathbb{R}_{\geq 0}$, $u(t) \in \mathcal{U}$ denotes the controllable input from the input set $\mathcal{U} \subset \mathbb{R}^{n_u}$, and $w(t) \in \mathcal{W}$ is the uncontrollable disturbance from the disturbance set $\mathcal{W} \subset \mathbb{R}^{n_w}$. Further, $\mathcal{X}^{(0)} \subset \mathbb{R}^{n_x}$ denotes the set of initial states, i.e., $x(0) \in \mathcal{X}^{(0)}$. ■

Definition 2 (Reachable Set): For a system as given in Definition 1, we define the reachable set as

$$\mathcal{R}_x^{(e)}(t) = \big\{x(t) \mid \forall \tilde{t} \in [0, t] : \dot{x}\left(\tilde{t}\right) = f\left(x\left(\tilde{t}\right), u\left(\tilde{t}\right), w\left(\tilde{t}\right)\right), x(0) \in \mathcal{X}^{(0)}, u\left(\tilde{t}\right) \in \mathcal{U}, w\left(\tilde{t}\right) \in \mathcal{W}\big\}.$$

■

Since in general, computing the exact reachable set $\mathcal{R}_x^{(e)}$ is not possible [53], we use overapproximative reachable sets $\mathcal{R}_x$ to ensure formal correctness and approximative reachable sets $\tilde{\mathcal{R}}_x$ for the iterative optimization of the controller. For easier readability, we do not explicitly state the dependence of $\mathcal{R}_x^{(e)}$, $\mathcal{R}_x$, or $\tilde{\mathcal{R}}_x$ on the initial set $\mathcal{X}^{(0)}$, input set $\mathcal{U}$, and disturbance set $\mathcal{W}$.

We require different set representations. For convenience, we first describe the concept of a generating function to generate a set. Then, we introduce zonotopes as a popular set representation for reachability analysis and polynomial zonotopes [54], [55] to represent non-convex sets. Subsequently, we define ellipsoids, which are helpful due to their compact representation size, and H-polytopes to intuitively represent constraints. Lastly, we introduce support functions, which enable us to easily extend a set in a given direction.

Definition 3 (Set Generation): We define

$$\{s(\Lambda)\}_\Lambda = \left\{s(\Lambda) \mid \Lambda \in [-1, 1]^{p\times m}\right\} = \mathcal{S},$$

where $s : [-1, 1]^{p\times m} \mapsto \mathbb{R}^n$ is the generating function of $\mathcal{S} \subset \mathbb{R}^n$ and $\Lambda \in [-1, 1]^{p\times m}$ are the dependent factors of $\mathcal{S}$. We say that $\mathcal{S}$ is generated by $s(\Lambda)$ over Λ. ■

Definition 4 (Zonotope): A zonotope $\mathcal{Z} = \langle c, G\rangle_Z$ with center $c \in \mathbb{R}^n$ and generator matrix $G \in \mathbb{R}^{n\times m}$ is given by

$$\mathcal{Z} = \{c + Gv\}_v,$$

where $c + Gv$ is its generating function with dependent factors $v \in [-1, 1]^m$. ■

Definition 5 (Polynomial Zonotope): Let $c \in \mathbb{R}^n$ be the starting point, $G = \left[g^{(1)}, \ \ldots, \ g^{(m)}\right] \in \mathbb{R}^{n\times m}$ the generator matrix with generators $g^{(i)} \in \mathbb{R}^n$ for $i \in \{1, \ldots, m\}$, and $E = \left[e^{(1)}, \ \ldots, \ e^{(m)}\right] \in \mathbb{N}^{d\times m}$ the exponent matrix of a polynomial zonotope. Its generating function is defined as

$$h(v) = c + \sum_{i=1}^{m} g^{(i)} v^{e^{(i)}},$$

with dependent factors $v \in [-1, 1]^d$ and where we used multi-index notation defined in Section II-A, so that we can construct a polynomial zonotope as $\mathcal{PZ} = \{h(v)\}_v$. ■

Definition 6 (Ellipsoid): An ellipsoid $\mathcal{E} = \langle c, Q\rangle_E$ with center $c \in \mathbb{R}^n$ and shape matrix $Q \in \mathbb{S}_{++}^{n\times n}$ is defined by

$$\mathcal{E} = \left\{x \in \mathbb{R}^n \mid (x - c)^T Q^{-1} (x - c) \leq 1\right\}.$$

■

We only consider non-degenerate ellipsoids, i.e., Q^{-1} exists, in this article. For a given matrix $M \in \mathbb{R}^{m\times n}$, the linear map of an ellipsoid is [56]

$$M \langle c, Q\rangle_E = \langle Mc, MQM^T\rangle_E .$$

Definition 7 (H-Polytope): An n-dimensional H-polytope $\langle C, d\rangle_H$ with the matrix $C \in \mathbb{R}^{m\times n}$ and offset $d \in \mathbb{R}^m$ is given by

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

■

Definition 8 (Support Function): The support function of a convex set $\mathcal{M} \subseteq \mathbb{R}^n$ in direction $l \in \mathbb{R}^n$ is defined as

$$\rho_{\mathcal{M}}(l) = \sup_{x\in\mathcal{M}} l^T x.$$

■

We define the shorthand $\rho_{\mathcal{M}}(L) = \left[\rho_{\mathcal{M}}\left(l^{(1)}\right), \ \ldots, \ \rho_{\mathcal{M}}\left(l^{(o)}\right)\right]^T$, where $L^T = \left[l^{(1)}, \ \ldots, \ l^{(o)}\right] \in \mathbb{R}^{n\times o}$. In this article, sets will often be parameterized, e.g., $\mathcal{M}(z) \subseteq \mathbb{R}^n$ for $z \in \mathbb{R}^m$. The corresponding support function will append these arguments, i.e., the support function for $\mathcal{M}(z)$ is then given by $\rho_{\mathcal{M}}(l, z)$. We are now ready to formulate the problem statement and propose our solution concept.

III. PROBLEM STATEMENT AND SOLUTION CONCEPT

For the remainder of this article, we assume that the initial set $\mathcal{X}^{(0)}$ and disturbance set $\mathcal{W}$ are zonotopes. We make no assumptions about the statistical nature of $\mathcal{W}$ but take $\mathcal{W}$ to be centered at 0; the system dynamics can always absorb any non-zero center.

We want to synthesize a controller $u(t, x(t))$ – dependent on time $t \in \mathbb{R}_{\geq 0}$ and state $x(t) \in \mathbb{R}^{n_x}$ – that steers a set of initial states $\mathcal{X}^{(0)} \subset \mathbb{R}^{n_x}$ as close as possible to a target state $x_\mathrm{f} \in \mathbb{R}^{n_x}$ within time $t_\mathrm{f} \in \mathbb{R}_+$ while bounded input constraints $\mathcal{U} \subset \mathbb{R}^{n_u}$ and (possibly unbounded) state constraints $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, both given as H-polytopes, are to be respected, i.e.

$$\min_{u(\cdot,x(\cdot))} \max_{x(t_\mathrm{f})} \|x\left(t_\mathrm{f}\right) - x_\mathrm{f}\|_1 , \tag{1a}$$

$$\text{s.t. } \forall t \in [0, t_\mathrm{f}] : \dot{x}(t) = f\left(x(t), u(t, x(t)), w(t)\right), \tag{1b}$$

$$\forall t \in [0, t_\mathrm{f}] : u\left(t, x(t)\right) \in \mathcal{U}, \tag{1c}$$

$$\forall t \in [0, t_\mathrm{f}] : x(t) \in \mathcal{X}. \tag{1d}$$

To obtain a tractable optimization problem, we parameterize the controller for each time interval $t \in \tau^{(i)} = [i, i+1]\frac{t_\mathrm{f}}{m}$, $0 \leq i \leq m-1$, as in [51] so that

$$u(t, x, P, K) = u_\mathrm{ff}\left(x(0), P^{(i)}\right) + K^{(i)}\left(x(t, P, K) - x_\mathrm{ff}(t, P)\right), \tag{2}$$

where $m \in \mathbb{N}_+$ is the number of piecewise constant feedforward and feedback controller pairs, $P^{(i)} \in [-1, 1]^{n_u\times a}$ are the feedforward control parameters ($a \in \mathbb{N}_+$ is the number of feedforward control parameters per input dimension), $P = \left[P^{(0)^T}, \ \ldots, \ P^{(m-1)^T}\right]^T$ collects all m feedforward parameter matrices, $K(t) = K^{(i)} \in \mathbb{R}^{n_u\times n_x}$ for $t \in \tau^{(i)}$ are the feedback gain matrices, and $x_\mathrm{ff}(t, P)$ is defined as the solution of

$$\dot{x}_\mathrm{ff}(t, P) = f\left(x_\mathrm{ff}(t, P), u_\mathrm{ff}\left(x(0), P^{(i)}\right), 0\right), \tag{3}$$

for $t \in \tau^{(i)}$ and $x(0) = x_\mathrm{ff}(0, P) \in \mathcal{X}^{(0)}$. We often omit indices and time dependencies where clear from context for readability, e.g., we write K instead of $K^{(i)}$. The feedforward state x_ff can be interpreted as the state of the undisturbed system and the proposed controller in (2) aims to follow the feedforward state as closely as possible by bringing the deviation vector $\Delta x(t, P, K)$, implicitly defined by

$$x(t, P, K) = x_\mathrm{ff}(t, P) + \Delta x(t, P, K), \tag{4}$$

as close to zero as possible, where $x(t, P, K) \in \mathcal{R}_x(t, P, K)$, with $\mathcal{R}_x(t, P, K)$ being an overapproximative, closed-loop reachable set of the disturbed flow using the controller in (2). We denote by $\mathcal{R}_{x_\mathrm{ff}}(t, P)$ the feedforward reachable set from the flow in (3) containing all $x_\mathrm{ff}(t, P)$ and construct $\mathcal{S}_{\Delta x}(t, P, K)$ to contain all state deviations $\Delta x(t, P, K)$ as defined in (4). When interpreting (4) in a set-based manner, we accept some conservatism and replace the exact sum by the Minkowski sum – neglecting dependencies between sets – which yields

$$x(t, P, K) \in \mathcal{R}_x(t, P, K) \subseteq \mathcal{R}_{x_\mathrm{ff}}(t, P) \oplus \mathcal{S}_{\Delta x}(t, P, K).$$

As shown later in Section VII, the added conservatism is negligible because both sets are only loosely coupled by the initial state. Let $\Delta u(t, P, K) = K(t)\Delta x(t, P, K) \in \mathcal{S}_{\Delta u}(t, P, K)$, where $\mathcal{S}_{\Delta u}(t, P, K)$ denotes the set of input deviations. We can then similarly write

$$u(t, x, P, K) \in \mathcal{S}_u(t, P, K) \subseteq \mathcal{S}_{u_\mathrm{ff}}(t, P) \oplus \mathcal{S}_{\Delta u}(t, P, K),$$

where $\mathcal{S}_u(t, P, K)$ and $\mathcal{S}_{u_\mathrm{ff}}(t, P)$ contain all combined inputs $u(t, x, P, K)$ and feedforward inputs $u_\mathrm{ff}\left(x(0), P^{(i)}\right)$ for $t \in \tau^{(i)}$, respectively. With these simplifications, (1) is relaxed to

$$\hat{P}, \hat{K} = \arg\min_{P,K} \left\{ \max_{x_\mathrm{ff}(t_\mathrm{f},P)} \|x_\mathrm{ff}\left(t_\mathrm{f}, P\right) - x_\mathrm{f}\|_1 + \max_{\Delta x(t_\mathrm{f},P,K)} \|\Delta x\left(t_\mathrm{f}, P, K\right)\|_1 \right\}, \tag{5a}$$

$$\text{s.t. } \forall t \in [0, t_\mathrm{f}] : (3),\ (4), \tag{5b}$$

$$\forall t \in [0, t_\mathrm{f}] : \mathcal{S}_{u_\mathrm{ff}}(t, P) \oplus \mathcal{S}_{\Delta u}(t, P, K) \subseteq \mathcal{U}, \tag{5c}$$

$$\forall t \in [0, t_\mathrm{f}] : \mathcal{R}_{x_\mathrm{ff}}(t, P) \oplus \mathcal{S}_{\Delta x}(t, P, K) \subseteq \mathcal{X}. \tag{5d}$$

Even though (5) is now easier to solve than (1) due to the aforementioned simplifications, it still requires reachable sets to guarantee formal correctness. Since this is computationally expensive, we instead solve an approximation of (5) by finding approximations to all required sets, followed by overapproximative reachability analysis to ensure constraint satisfaction. We illustrate our solution concept with Example 1 visualized in Fig. 1, where we omit the feedback

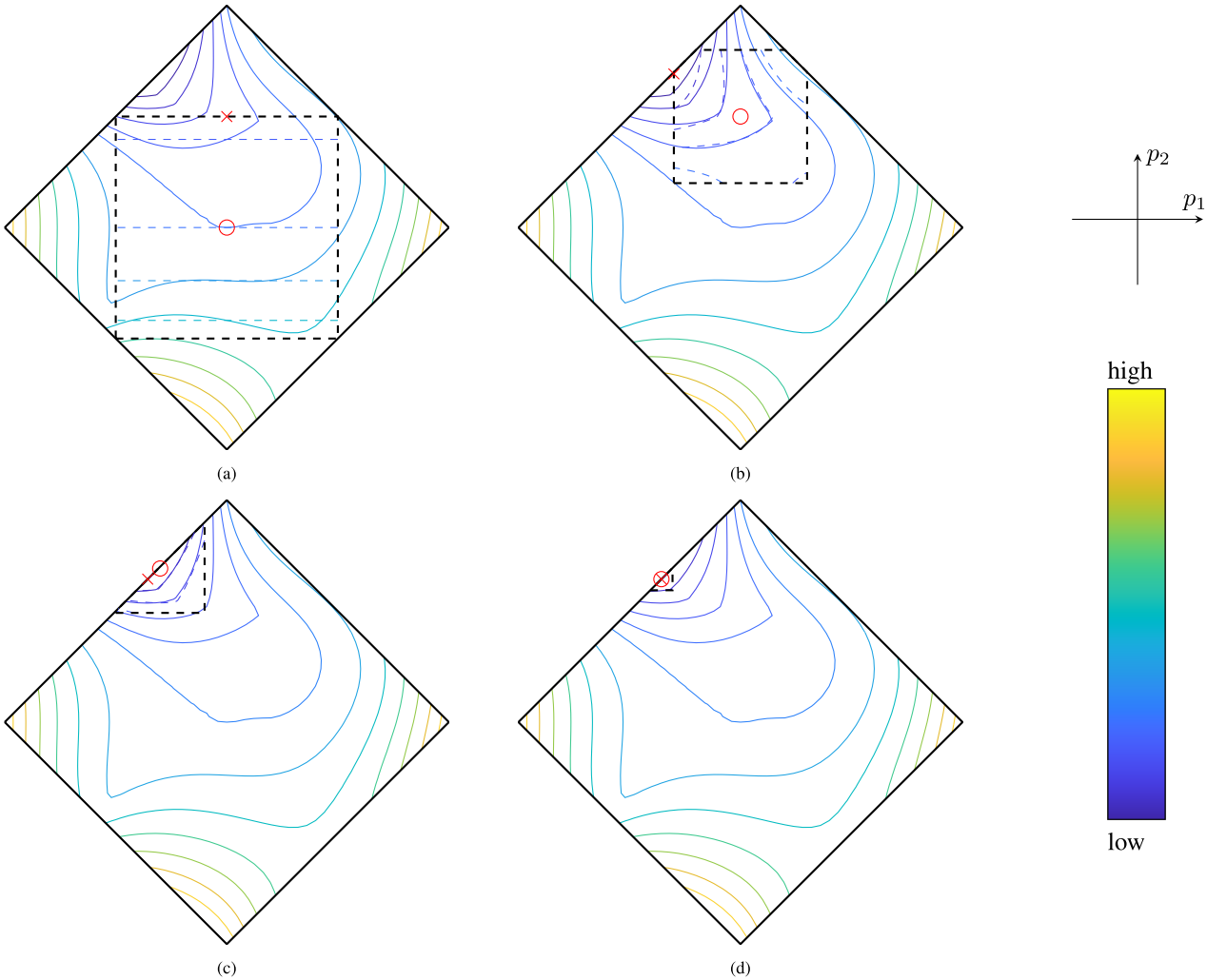

FIGURE 1. Visualization of our approach for Example 1 over four iterations. Shown are input constraints (outer black diamond), the contour lines of the controller cost $J(P)$ (solid) and the approximated cost $\tilde{J}(P)$ (dashed) for the current trust region (dashed black box), the current initial guess of the control parameters $\bar{P}$ (red circle), and the optimizer $\hat{P}$ of the approximated optimization problem (red x).

controller and limit ourselves to a single feedforward controller, i.e., $m = 1$, subject to input constraints for illustrative purposes.

Example 1: Let the zonotope overapproximation [55, Prop. 5] of the final, closed-loop reachable set $\mathcal{R}_x(t_\mathrm{f}, P)$ of a 1-dimensional system be given by

$$\mathcal{Z}_x(t_\mathrm{f}, P) = \left\langle -p_2 + 2p_2^2 + 4p_1^3, \begin{bmatrix} 3 + 10p_1p_2^3, & 1 - p_2 \end{bmatrix} \right\rangle_Z,$$

where $\bar{u}_\mathrm{ff}(\beta, P) = p_1 + p_2\beta$ is the controller template (see Section IV-A for details), $\beta \in [-1, 1]$ denotes the dependent factor of the initial set $\mathcal{X}^{(0)}$, $P = P^{(0)} = [p_1, \quad p_2]$, $x_\mathrm{f} = 0$, and $\mathcal{U} = [-1, 1] = \{\tilde{u}_\mathrm{ff} \in \mathbb{R} \mid A_\mathcal{U}\tilde{u}_\mathrm{ff} \leq b_\mathcal{U}\}$ with $A_\mathcal{U} = [1, \quad -1]^T$ and $b_\mathcal{U} = [1, \quad 1]^T$. Since $\bar{u}_\mathrm{ff}(\beta, P)$ is linear in β, we have $\forall t \in [0, t_\mathrm{f}] : \mathcal{S}_{u_\mathrm{ff}}(t, P) = \mathcal{Z}_{u_\mathrm{ff}}(t, P) = \langle p_1, p_2 \rangle_Z$ (see (16)). The controller cost (see Section VI-B for details) is

$$J(P) = \left|-p_2 + 2p_2^2 + 4p_1^3\right| + \left|3 + 10p_1p_2^3\right| + |1 - p_2| + 100\max(0, |p_1| + |p_2| - 1).$$

Fig. 1 shows the contour lines of $J(P)$.

For the first iteration, we start at $\bar{P} = 0$ and initially set the trust-region radius $\gamma \in (0, 1]$, which constrains the available set of feedforward parameters (see Section IV-B for details), to $\gamma = \frac{1}{4}$, i.e., $\mathcal{P}_\gamma(\bar{P}) = \left[-\frac{1}{2}, \frac{1}{2}\right]$ (see (10)). Fig. 1(a) shows the contour lines for our approximation $\tilde{J}(P)$ of $J(P)$ in the first iteration. Clearly, the approximation $\tilde{J}(P)$ is not very accurate even though $J(\hat{P}) < J(\bar{P})$, and so we accept the step but shrink the trust-region radius γ for the next iteration. After another two accepted iterations (see Fig. 1(b) and (c)), we eventually arrive at a local minimum of $J(P)$ in the fourth iteration (see Fig. 1(d)) and thus terminate the algorithm. ■

Since we can only accurately approximate the controller cost locally, we use the trust-region radius to restrict the domain of our approximation to a trust region, i.e., a bounded region of our control parameters for which we trust the approximation to be accurate.

The remainder of this article is structured as follows: We derive the set of feedforward inputs $\mathcal{S}_{u_\mathrm{ff}}(t, P)$ and the

approximation for the feedforward reachable set $\mathcal{R}_{x_{\text{ff}}}(t, P)$ in Section IV. Then, we compute approximations to the state deviation set $\mathcal{S}_{\Delta x}(t, P, K)$ and the input deviation set $\mathcal{S}_{\Delta u}(t, P, K)$ in Section V. In Section VI, we formulate an approximation to (5) using findings from Sections IV and V, show how we can control the accuracy of this approximation using trust regions, and derive the computational complexity of the proposed algorithm. Finally, we demonstrate the applicability of our novel approach using numerical benchmarks in Section VII.

IV. FEEDFORWARD REACHABLE SET

To compute an approximation of the feedforward reachable set, we first introduce the parameterization of the feedforward controller $u_{\text{ff}}\left(x(0), P^{(i)}\right)$ in Section IV-A. We then discuss the feedforward reachable set computation using the parameterized controller in Section IV-B.

A. CONTROLLER TEMPLATE

As defined in Section III, the feedforward controller is state-dependent, i.e., we compute a feedforward control law for each initial state $x(0) \in \mathcal{X}^{(0)} = \{c + G\beta\}_\beta$ with center $c \in \mathbb{R}^{n_x}$, generator matrix $G \in \mathbb{R}^{n_x \times l}$, and dependent factors $\beta \in [-1, 1]^l$. For numerical purposes, it will be beneficial to parameterize the initial set in its dependent factors β. Let us define $E = \left[e^{(1)}, \quad \ldots, \quad e^{(a)}\right] \in \mathbb{N}^{l \times a}$ as a matrix of exponents and $P^{(i)} = \left[p^{(i,1)}, \quad \ldots, \quad p^{(i,a)}\right] \in \mathbb{R}^{n_u \times a}$ as the matrix of feedforward parameters for $0 \leq i \leq m-1$. In [52], the parameterization

$$\bar{u}_{\text{ff}}\left(\beta, P^{(i)}\right) = c_{\bar{\mathcal{U}}} + G_{\bar{\mathcal{U}}}\left(\sum_{k=1}^{a} p^{(i,k)} \beta^{e^{(k)}}\right), \tag{6}$$

is proposed, where $\bar{\mathcal{U}} = \left\{c_{\bar{\mathcal{U}}} + G_{\bar{\mathcal{U}}}\alpha\right\}_\alpha$ is the input set overapproximated by a parallelotope to achieve uniform scaling: By comparing (6) with $\bar{u}(\alpha) = c_{\bar{\mathcal{U}}} + G_{\bar{\mathcal{U}}}\alpha$, we can parameterize α instead of the input directly. Further, we have $\mathcal{U} \subseteq \bar{\mathcal{U}} \subseteq \left\{\bar{u}_{\text{ff}}\left(\beta, P^{(i)}\right)\right\}_{\beta, P^{(i)}}$ since $[-1, 1]^{n_u} \subseteq \left\{\sum_{k=1}^{a} p^{(i,k)} \beta^{e^{(k)}}\right\}_{\beta, P^{(i)}}$, and therefore $\forall \beta \in [-1, 1]^l : \bar{u}_{\text{ff}}\left(\beta, P^{(i)}\right) \in \mathcal{U} \Rightarrow P^{(i)} \in [-1, 1]^{n_u \times a}$, which ensures uniform scaling of $P^{(i)}$, i.e., the absolute magnitude of each element is less or equal to one. When $\mathcal{U}$ is a zonotope, the parallelotope enclosure can be achieved through order reduction [57]. A halfspace representation of $\mathcal{U}$, as assumed in this article, can be enclosed as follows:

Proposition 1 (H-Polytope to Parallelotope): Let $\mathcal{M} = \langle A, b\rangle_H$ with $A \in \mathbb{R}^{o \times n}$ and $b \in \mathbb{R}^o$ be a bounded polytope. Then a parallelotope overapproximation of $\mathcal{M}$ is given by $\mathcal{Z} = Q^{\frac{1}{2}} \langle \tilde{c}, \text{diag}(\tilde{r})\rangle_Z \oplus \{c\}$, where

$$\tilde{c} = \frac{1}{2}\left(\rho_{\tilde{\mathcal{M}}}(I) + \rho_{\tilde{\mathcal{M}}}(-I)\right),$$

$$\tilde{r} = \frac{1}{2}\left(\rho_{\tilde{\mathcal{M}}}(I) - \rho_{\tilde{\mathcal{M}}}(-I)\right),$$

$$\tilde{\mathcal{M}} = \left\langle AQ^{\frac{1}{2}}, b - Ac\right\rangle_H,$$

and where $\langle c, Q\rangle_E$ with center $c \in \mathbb{R}^n$ and shape matrix $Q \in \mathbb{S}_{++}^{n \times n}$ is the maximum-volume ellipsoid inscribed into $\mathcal{M}$ [58, Sec. 8.4.2].

Proof: We shift $\mathcal{M}$ by c so that $0 \in \mathcal{M} \oplus \{-c\}$ and use the shape matrix Q to transform $\mathcal{M}$ into roughly a hypercube, i.e., $\tilde{\mathcal{M}} = Q^{-\frac{1}{2}}(\mathcal{M} \oplus \{-c\}) = \left\langle AQ^{\frac{1}{2}}, b - Ac\right\rangle_H$, since $Q^{-\frac{1}{2}}(\langle c, Q\rangle_E \oplus \{-c\}) = \langle 0, I_n\rangle_E$. A parallelotope enclosing $\tilde{\mathcal{M}}$ is then given by $\tilde{\mathcal{Z}} = \langle \tilde{c}, \text{diag}(\tilde{r})\rangle_Z = \{\tilde{z} \in \mathbb{R}^n \mid \rho_{\tilde{\mathcal{M}}}(-I) \leq \tilde{z} \leq \rho_{\tilde{\mathcal{M}}}(I)\}$ due to the definition of the support function, where $\tilde{c}$ and $\tilde{r}$ are given as above. Applying the inverse transform and shifting the result by c yields $\mathcal{Z} = Q^{\frac{1}{2}}\tilde{\mathcal{Z}} \oplus \{c\}$ which concludes the proof. ■

Since computing the maximum-volume ellipsoid of $\mathcal{U}$ can be posed as a semi-definite programming (SDP) problem [58, Sec. 8.4.2], Proposition 1 can be efficiently solved. The computed overapproximation $\bar{\mathcal{U}}$ is only required for the proper scaling of $P^{(i)}$; the original input set $\mathcal{U}$ is used to verify input constraint satisfaction.

If $\mathcal{X}^{(0)}$ is given as a non-degenerate parallelotope, i.e., G^{-1} exists, the controller template for the state $x(0) = c + G\beta \in \mathcal{X}^{(0)}$ is given by

$$u_{\text{ff}}\left(x(0), P^{(i)}\right) = \bar{u}_{\text{ff}}\left(G^{-1}\left(x(0) - c\right), P^{(i)}\right). \tag{7}$$

If $\mathcal{X}^{(0)}$ is a zonotope, the dependent factors β for a given $x(0)$ can be obtained by solving $x(0) = c + G\beta$ for $\|\beta\|_\infty \leq 1$ using linear programming. The exact set of possible feedforward inputs is given by

$$\mathcal{S}_{u_{\text{ff}}}(t, P) = \left\{\bar{u}_{\text{ff}}\left(\beta, P^{(i)}\right)\right\}_\beta, \; t \in \tau^{(i)}, \; 0 \leq i \leq m-1. \tag{8}$$

B. REACHABLE SET COMPUTATION

To efficiently solve (5), we approximate the feedforward reachable set so that it is obtained by the evaluation of a polynomial map without the need to recompute the reachable set for every given P.

We can construct such a polynomial map using polynomial zonotopes [55] and an extended system state as follows: Given the extended state $x_{\text{ext}} = \left[x_{\text{ff}}^T, \quad u_{\text{ff}}^T\right]^T$, which is necessary to retain the dependency of both the initial state as well as the input on β, the extended initial set is

$$\mathcal{X}_{\text{ext}}^{(0)}\left(P^{(0)}\right) = \left\{\begin{bmatrix} x^{(0)}(\beta) \\ u_{\text{ff}}\left(x^{(0)}(\beta), P^{(0)}\right)\end{bmatrix}\right\}_\beta,$$

where $x^{(0)}(\beta) = c + G\beta$ is the generating function of $\mathcal{X}^{(0)}$. Applying reachability analysis as described in [55] eventually yields

$$\hat{\mathcal{R}}_{x_{\text{ff}}}(t, P) = \mathcal{D}(t, P) \oplus \langle c_{\text{err}}, G_{\text{err}}\rangle_Z, \tag{9}$$

for $t \in [0, t_{\text{f}}]$ and $P \in \mathcal{P} = [-1, 1]^{mn_u \times a}$, where $\mathcal{D}(t, P)$ is the part of $\hat{\mathcal{R}}_{x_{\text{ff}}}(t, P)$ that retained its dependence on P and $\langle c_{\text{err}}, G_{\text{err}}\rangle_Z$ is the zonotope bounding abstraction and reduction errors. However, the size of the error $\langle c_{\text{err}}, G_{\text{err}}\rangle_Z$ depends on the size of $\mathcal{P}$ as demonstrated next.

Example 2: Let us consider a controlled van-der-Pol oscillator with the undisturbed dynamics $f(x, u) =$

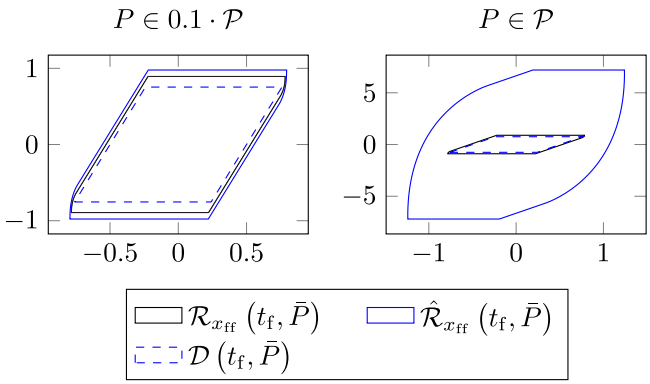

FIGURE 2. Approximation of the tight overapproximative reachable set $\mathcal{R}_{x_{\mathrm{ff}}}(t_\mathrm{f}, P)$ by $\hat{\mathcal{R}}_{x_{\mathrm{ff}}}(t_\mathrm{f}, P) = \mathcal{D}(t_\mathrm{f}, P) \oplus \langle c_{\mathrm{err}}, G_{\mathrm{err}} \rangle_Z$ for $P \in 0.1 \cdot \mathcal{P}$ and for $P \in \mathcal{P} = [-1, 1]^{mn_u \times a}$, evaluated at $\bar{P} = 0.1 \cdot 1_{mn_u \times a}$.

$\left[x_2, \quad \left(1 - x_1^2\right) x_2 - x_1 + u\right]^T$, $\mathcal{X}^{(0)} = \left\{ \begin{bmatrix} 0.5\beta_1 \\ 0.3\beta_2 \end{bmatrix} \right\}_\beta$, $t_\mathrm{f} = 0.5$ s, $\bar{u}_{\mathrm{ff}}\left(\beta, P^{(i)}\right) = 5\left(p^{(i,1)}\beta_1 + p^{(i,2)}\beta_2\right)$ with $i \in \{1, \ldots, 5\}$, $P^{(i)} = \left[p^{(i,1)}, \quad p^{(i,2)}\right] \in [-1, 1]^{1 \times 2}$, and $E = I_2$. We compute $\hat{\mathcal{R}}_{x_{\mathrm{ff}}}(t_\mathrm{f}, P)$ as described above. Fig. 2 visualizes the reachable sets $\hat{\mathcal{R}}_{x_{\mathrm{ff}}}(t_\mathrm{f}, P)|_{P=\bar{P}}$ for $\bar{P} = 0.1 \cdot 1_{mn_u \times a}$, computed for both $P \in 0.1 \cdot \mathcal{P}$ and $P \in \mathcal{P}$, and compares them to the overapproximative reachable set $\mathcal{R}_{x_{\mathrm{ff}}}(t_\mathrm{f}, \bar{P})$, which needs to be recomputed for each $\bar{P}$. Evidently, choosing P from a larger set affects the accuracy of the parameterized reachable set. For both set sizes, however, we notice that $\mathcal{D}(t_\mathrm{f}, \bar{P})$ approximates $\mathcal{R}_{x_{\mathrm{ff}}}(t_\mathrm{f}, \bar{P})$ reasonably well, even if $\langle c_{\mathrm{err}}, G_{\mathrm{err}} \rangle_Z$ is much larger for $P \in \mathcal{P}$. ■

Thus, we avoid a large input set by computing $\hat{\mathcal{R}}_{x_{\mathrm{ff}}}(t, P)$ only for $P \in \mathcal{P}_\gamma(\bar{P})$ where

$$\mathcal{P}_\gamma\left(\bar{P}\right) = \left\{ \bar{P} + 2\gamma M \;\middle|\; M \in [-1, 1]^{mn_u \times a} \right\}, \tag{10}$$

and the trust-region radius $\gamma \in (0, 1]$ is reduced until the approximation is accurate enough. We include the factor 2 in (10) so that $\forall \bar{P} \in \mathcal{P} : \mathcal{P} \subseteq \mathcal{P}_1(\bar{P})$. Because we compute overapproximative reachability analysis after each trust-region iteration to formally verify constraint satisfaction, we have $\mathcal{R}_{x_{\mathrm{ff}}}(t, \bar{P})$ at the previously computed feedforward control parameters $\bar{P}$ available (see Section VI-B for details). Therefore, we approximate the effect of the deviation from $\bar{P}$ by defining the exact sum $\oplus_e$ of polynomial zonotopes [55, Prop. 10] preserving dependencies and only considering deviations from $\mathcal{R}_{x_{\mathrm{ff}}}(t, \bar{P})$, where we approximate the relative change using $\hat{\mathcal{R}}_{x_{\mathrm{ff}}}(t, P)$, i.e., we define an approximation to the parameterized reachable set by

$$\begin{aligned} \tilde{\mathcal{R}}_{x_{\mathrm{ff}}}(t, P) &= \mathcal{R}_{x_{\mathrm{ff}}}\left(t, \bar{P}\right) \oplus_e \hat{\mathcal{R}}_{x_{\mathrm{ff}}}(t, P) \oplus_e \left(-\hat{\mathcal{R}}_{x_{\mathrm{ff}}}\left(t, \bar{P}\right)\right) \\ &= \mathcal{R}_{x_{\mathrm{ff}}}\left(t, \bar{P}\right) \oplus_e \mathcal{D}(t, P) \oplus_e \left(-\mathcal{D}\left(t, \bar{P}\right)\right), \end{aligned} \tag{11}$$

for $P \in \mathcal{P}_\gamma(\bar{P})$ and where the last equality follows from $\hat{\mathcal{R}}_{x_{\mathrm{ff}}}(t, P) \oplus_e \left(-\hat{\mathcal{R}}_{x_{\mathrm{ff}}}\left(t, \bar{P}\right)\right) = \mathcal{D}(t, P) \oplus_e \left(-\mathcal{D}\left(t, \bar{P}\right)\right)$ (see (9)).

V. DISTURBANCE SET COMPUTATION

To solve (5), we also require approximations to the state deviation set $\mathcal{S}_{\Delta x}(t, P, K)$ and input deviation set $\mathcal{S}_{\Delta u}(t, P, K)$, which we derive in Section V-A. In Section V-B, we then briefly describe a parameterization of the gain matrices $K^{(i)}$ for $0 \leq i \leq m - 1$ using linear-quadratic regulator (LQR) control from [51, Sec. IV.B.] to reduce the number of optimization variables.

A. REACHABLE SET COMPUTATION

For now, we assume that all gain matrices $K(t, z) = K^{(i)}(z)$ for $t \in \tau^{(i)}$ with $0 \leq i \leq m - 1$ are parameterized in two matrices Q and R as well as the feedforward parameters P, collected in the vector $z = \left[P_{(:)}^T, \quad Q_{(:)}^T, \quad R_{(:)}^T\right]^T$, where Q and R as well as the parameterization itself will be introduced in Section V-B.

To compute an approximation of the disturbance tube, we find an approximate flow equation for the state deviation Δx by a first-order Taylor expansion of Δx around $x = x_{\mathrm{ff}}$, $u = u_{\mathrm{ff}}$, and $w = 0$, i.e.

$$\begin{aligned} \Delta\dot{x} &= \dot{x} - \dot{x}_{\mathrm{ff}} = f(x, u, w) - f\left(x_{\mathrm{ff}}, u_{\mathrm{ff}}, 0\right) \\ &\approx \left(A(t, P) + B(t, P)K(t, z)\right)\Delta x + V(t, P)w, \end{aligned} \tag{12}$$

with

$$A(t, P) = \left.\frac{\partial f(x, u, w)}{\partial x}\right|_{\substack{x = \bar{x}(t, P) \\ u = \bar{u}(t, P) \\ w = 0}}, \tag{13}$$

$$B(t, P) = \left.\frac{\partial f(x, u, w)}{\partial u}\right|_{\substack{x = \bar{x}(t, P) \\ u = \bar{u}(t, P) \\ w = 0}}, \tag{14}$$

$$V(t, P) = \left.\frac{\partial f(x, u, w)}{\partial w}\right|_{\substack{x = \bar{x}(t, P) \\ u = \bar{u}(t, P) \\ w = 0}}, \tag{15}$$

where $\bar{x}(t, P) = \frac{1}{2}\left(\tilde{c}_{x_{\mathrm{ff}}}\left(\underline{t}, P\right) + \tilde{c}_{x_{\mathrm{ff}}}\left(\bar{t}, P\right)\right)$ and $\bar{u}(t, P) = c_{u_{\mathrm{ff}}}\left(\frac{1}{2}\left(\underline{t} + \bar{t}\right), P\right)$ are kept constant for $t \in \left[\underline{t}, \bar{t}\right] = [k, k + 1]\delta$ with $\delta = \frac{t_\mathrm{f}}{mq}$ and $0 \leq k \leq mq - 1$, so that we obtain mq linear time-invariant (LTI) systems for $q \in \mathbb{N}_+$. Here, we use the centers of the zonotope overapproximations [55, Prop. 5]

$$\mathcal{Z}_{u_{\mathrm{ff}}}(t, P) = \left\langle c_{u_{\mathrm{ff}}}(t, P), G_{u_{\mathrm{ff}}}(t, P) \right\rangle_Z, \tag{16}$$

$$\tilde{\mathcal{Z}}_{x_{\mathrm{ff}}}(t, P) = \left\langle \tilde{c}_{x_{\mathrm{ff}}}(t, P), \tilde{G}_{x_{\mathrm{ff}}}(t, P) \right\rangle_Z, \tag{17}$$

of $\mathcal{S}_{u_{\mathrm{ff}}}(t, P)$ and $\tilde{\mathcal{R}}_{x_{\mathrm{ff}}}(t, P)$ from (8) and (11) as an approximation of their geometric centers since the starting point of a polynomial zonotope (see Definition 5) is not a good approximation of its geometric center due to its polynomial dependent factors. Furthermore, we require both zonotope overapproximations in Section VI-A since zonotopes make an efficient evaluation of their support function possible [59].

We remark here that we do not assume the same linearized dynamics for the complete time horizon $[0, t_f]$ but rather compute a new linearization for each time interval $t \in [k, k+1]\delta$. A similar approach which uses (12) to approximate the deviation vector Δx for robustification of an optimal control problem is described in [60]. In general, approximation errors – caused by the first-order Taylor series and linearization – remain small due to the corrective actions of the controller so that the deviation from the feedforward solution stays small; that said, the linearized flow can become inaccurate for large disturbance sets $\mathcal{W}$. However, since we are not dependent on an exact approximation, approximation errors do not affect the soundness of our approach.

The reachable set $\tilde{\mathcal{Z}}_{\Delta x}(t, z)$ of the approximated flow (12) with $\tilde{\mathcal{Z}}_{\Delta x}(0, z) = \{0\}$ can be efficiently computed using zonotopes, as only Minkowski sums and linear maps are required, under which zonotopes are closed. Since the representation size of $\tilde{\mathcal{Z}}_{\Delta x}(t, z)$ can become quite large (due to computing many Minkowski sums), we can avoid its explicit computation by replacing all set constraints in (5) with support functions evaluated in the normal directions given by the H-representations of the polytopic input and polytopic state constraints (see Section VI-A for details; also [61]). As we use overapproximative reachability analysis for each trust-region iteration (see Section VI-B and Algorithm 1), the extended reachable set at $z = \bar{z}$ is available (see Section VI-B for details) and hence the unknown support function $\rho_{\Delta x}(l, t, z)$ of the zonotope overapproximation $\mathcal{Z}_{\Delta x}(t, z) \supseteq \mathcal{S}_{\Delta x}(t, z)$ in the direction $l \in \mathbb{R}^{n_x}$ can be efficiently computed at $z = \bar{z}$. We first define a rough approximation of this unknown support function as

$$\tilde{\rho}^{(\mathcal{E})}_{\Delta x}(l, t, z) = \frac{\rho_{\Delta x}(l, t, \bar{z})}{\rho^{(\mathcal{E})}_{\Delta x}(l, t, \bar{z})} \rho^{(\mathcal{E})}_{\Delta x}(l, t, z),$$

$$\rho^{(\mathcal{E})}_{\Delta x}(l, t, z) = \sqrt{l^T Q_{\Delta x}(t, z) l + \epsilon},$$

where $\epsilon \ll 1$ is used to achieve differentiability and we scale $\tilde{\rho}^{(\mathcal{E})}_{\Delta x}$ so that $\tilde{\rho}^{(\mathcal{E})}_{\Delta x}(l, t, \bar{z}) = \rho_{\Delta x}(l, t, \bar{z})$. Further, $\mathcal{E}_{\Delta x}(t, z) = \langle 0, Q_{\Delta x}(t, z) \rangle_E$ is obtained by computing the reachable set of the linearized flow (12) using the ellipsoid $\mathcal{E}_{\mathcal{W}} = \langle 0, Q_{\mathcal{W}} \rangle_E \supseteq \mathcal{W}$ for $Q_{\mathcal{W}} \in \mathbb{S}^{n_w \times n_w}_{++}$ (see, e.g., [62]). For $t + \delta \leq t_f$, we thus have [63]

$$\mathcal{E}_{\Delta x}(t + \delta, z) = e^{\bar{A}(t,z)\delta} \mathcal{E}_{\Delta x}(t, z) \oplus \int_0^\delta e^{\bar{A}(t,z)\phi} V(t, P) \mathcal{E}_{\mathcal{W}} \, \mathrm{d}\phi, \tag{18}$$

with $\bar{A}(t, z) = A(t, P) + B(t, P)K(t, z)$ and where $\mathcal{E}_{\Delta x}(0, z) = \{0\}$. We approximate the Minkowski sum of two ellipsoids required in (18) as the sum of their shape matrices. The approximation to the unknown support function $\rho_{\Delta x}(l, t, z)$ can then be defined as

$$\tilde{\rho}_{\Delta x}(l, t, z) = \rho_{\Delta x}(l, t, \bar{z}) + \tilde{\rho}^{(\mathcal{E})}_{\Delta x}(l, t, z) - \tilde{\rho}^{(\mathcal{E})}_{\Delta x}(l, t, \bar{z}), \tag{19}$$

so that we only approximate the unknown difference $\rho_{\Delta x}(l, t, z) - \rho_{\Delta x}(l, t, \bar{z})$ by $\tilde{\rho}^{(\mathcal{E})}_{\Delta x}(l, t, z) - \tilde{\rho}^{(\mathcal{E})}_{\Delta x}(l, t, \bar{z})$ since $\tilde{\rho}_{\Delta x}(l, t, \bar{z}) = \rho_{\Delta x}(l, t, \bar{z})$ holds for $z = \bar{z}$, which follows by substitution of $\bar{z}$ into (19). Furthermore, we have

$$\mathcal{E}_{\Delta u}(t, z) = K(t, z) \mathcal{E}_{\Delta x}(t, z),$$

since $u(t, x, z) = u_{\mathrm{ff}}\left(x(0), P^{(i)}\right) + K(t, z)\Delta x(t, z)$ for $t \in \tau^{(i)}$ with $0 \leq i \leq m - 1$, from which $\tilde{\rho}_{\Delta u}(l, t, z)$ can be derived analogously to $\tilde{\rho}_{\Delta x}(l, t, z)$. Next, we introduce the feedback matrix parameterization.

B. FEEDBACK MATRIX PARAMETERIZATION

Directly optimizing over all gain matrices $K^{(i)}$ for $0 \leq i \leq m - 1$ as in (5) requires $mn_u n_x$ optimization variables since $K^{(i)} \in \mathbb{R}^{n_u \times n_x}$. Therefore, we introduce a parameterization of the feedback matrices using LQR control [51, Sec. IV.B] to reduce the number of optimization variables.

With m feedback matrices to be computed, it makes sense to only consider those $K^{(i)}$ that asymptotically stabilize the m LTI systems with system matrices $A(t, P) + B(t, P)K^{(i)}$ and input matrices $V(t, P)$ as defined in (13) to (15) but which are now assumed to be constant in time for $t \in [\underline{t}, \bar{t}] = \tau^{(i)} = [i, i + 1]\frac{t_f}{m}$. For any such system and matrices $Q \in \mathbb{S}^{n_x \times n_x}_{++}$ and $R \in \mathbb{S}^{n_u \times n_u}_{++}$, LQR control only generates feedback matrices that result in an asymptotically stable system by design, assuming that all m LTI systems are controllable for all P. This motivates the parameterization $K(t, z) = K^{(i)}(z)$ for $t \in \tau^{(i)}$ with $z = \left[P^T_{(:)}, \quad Q^T_{(:)}, \quad R^T_{(:)}\right]^T$, so that optimization over K in (5) can be replaced by optimization over z. We refer the reader to the Appendix for a proof of the differentiability of K with respect to z. If controllability cannot be assumed, one can always directly optimize over K. Next, we introduce the novel trust-region synthesis approach.

VI. COMBINED SYNTHESIS

We now propose our novel trust-region approach for the combined synthesis problem. In Section VI-A, we construct a continuously differentiable optimization problem as an approximation to (5) using findings from Section IV and Section V, and then describe the proposed iterative algorithm in Section VI-B. In Section VI-C, we discuss the approximation accuracy of the trust-region subproblem, followed by a short description of the computational complexity of the algorithm in Section VI-D.

A. TRUST-REGION SUBPROBLEM

Given the current control parameters $\bar{P}$, $\bar{Q}$, and $\bar{R}$ – where $\rho_{u_{\mathrm{ff}}}$ is the support function of $\mathcal{Z}_{u_{\mathrm{ff}}}$ from (16) and $\tilde{\rho}_{\Delta x}$, $\tilde{\rho}_{\Delta u}$ are the approximated support functions as described in Section V-A – the solution of

$$\hat{z}, \hat{s} = \arg\min_{z,s} \tilde{J}_{\mathrm{TR}}(z, s), \tag{20a}$$

$$\text{s.t. } \tilde{g}_{\mathcal{U}}(z) \leq s_{\mathcal{U}}, \tag{20b}$$

$$\tilde{g}_{\mathcal{X}}(z) \leq s_{\mathcal{X}}, \tag{20c}$$

$$P \in \mathcal{P}_\gamma\left(\bar{P}\right) \cap \mathcal{P}, \tag{20d}$$

$$Q \in \left\{\bar{Q} + h_Q\left(\eta M\right)\right\}_M \cap \mathcal{Q}, \tag{20e}$$

$$R \in \left\{\bar{R} + h_R\left(\eta N\right)\right\}_N \cap \mathcal{R}, \tag{20f}$$

$$s = \begin{bmatrix} s_{\mathcal{U}}^T, & s_{\mathcal{X}}^T \end{bmatrix}^T \geq 0, \tag{20g}$$

with

$$\begin{aligned} \tilde{J}_{\mathrm{TR}}(z, s) = & \left\| \tilde{c}_{x_{\mathrm{ff}}}\left(t_{\mathrm{f}}, P\right) - x_{\mathrm{f}} \right\|_1 + \left\| \tilde{G}_{x_{\mathrm{ff}}}\left(t_{\mathrm{f}}, P\right)_{(:)} \right\|_1 \\ & + 1_{n_x}^T \tilde{\rho}_{\Delta x}\left(I_{n_x}, t_{\mathrm{f}}, z\right) + \sigma \left\| s \right\|_1, \end{aligned} \tag{21}$$

as well as

$$\tilde{g}_{\mathcal{U}}(z) = \max_{t \in [0, t_{\mathrm{f}}]} \left(\rho_{u_{\mathrm{ff}}}\left(A_{\mathcal{U}}, t, P\right) + \tilde{\rho}_{\Delta u}\left(A_{\mathcal{U}}, t, z\right)\right) - b_{\mathcal{U}}, \tag{22}$$

$$\tilde{g}_{\mathcal{X}}(z) = \max_{t \in [0, t_{\mathrm{f}}]} \left(\tilde{\rho}_{x_{\mathrm{ff}}}\left(A_{\mathcal{X}}, t, P\right) + \tilde{\rho}_{\Delta x}\left(A_{\mathcal{X}}, t, z\right)\right) - b_{\mathcal{X}}, \tag{23}$$

and critical point $(\hat{z}, \hat{s})$ is an approximation of (5) as explained subsequently:

- (20a) and (21): We replace $\mathcal{R}_{x_{\mathrm{ff}}}(t, P)$ with $\tilde{\mathcal{Z}}_{x_{\mathrm{ff}}}(t, P) = \left\langle \tilde{c}_{x_{\mathrm{ff}}}(t, P), \tilde{G}_{x_{\mathrm{ff}}}(t, P) \right\rangle_Z$ from (17) and construct (21) as follows: We penalize deviations from x_{f} (first term) and the size of its generators (second term). Further, we penalize the size of the disturbance tube by penalizing its support function values in the unit directions (third term) and introduce $\sigma \in \mathbb{R}_+$, which is chosen large enough (see, e.g., [64] for a discussion on the choice of σ) such that achieving feasibility is always prioritized over minimizing the objective value for $s = 0$ (fourth term).
- (20b): We have $\mathcal{A} \subseteq \mathcal{B} \iff \forall l \in \mathbb{R}^n : \rho_{\mathcal{A}}(l) \leq \rho_{\mathcal{B}}(l)$ for two convex sets $\mathcal{A} \subseteq \mathbb{R}^n$ and $\mathcal{B} \subseteq \mathbb{R}^n$, or $\mathcal{A} \subseteq \mathcal{B} \iff \rho_{\mathcal{A}}(C_{\mathcal{B}}) \leq d_{\mathcal{B}}$ if $\mathcal{B} = \langle C_{\mathcal{B}}, d_{\mathcal{B}} \rangle_H$ is an H-polytope. Thus, the set containment constraint in (5c) is encoded by the support function values in normal directions $A_{\mathcal{U}}^T$, where $\mathcal{U} = \langle A_{\mathcal{U}}, b_{\mathcal{U}} \rangle_H$. Here, we use $\rho_{\mathcal{M} \oplus \mathcal{N}}(l) = \rho_{\mathcal{M}}(l) + \rho_{\mathcal{N}}(l)$ for two convex sets $\mathcal{M} \subseteq \mathbb{R}^n$ and $\mathcal{N} \subseteq \mathbb{R}^n$ [65, Prop. 2.3]. Further, we relax the resulting input constraint $\tilde{g}_{\mathcal{U}}(z) \leq 0$ by adding $s_{\mathcal{U}} \geq 0$ to the right-hand side, which ensures that a feasible solution always exists, even if there is no feasible solution for the original constraint with $s_{\mathcal{U}} = 0$. Finally, the maximization of the support functions over t in $\tilde{g}_{\mathcal{U}}(z)$ can be approximated by evaluating (22) over smaller time intervals and forming a discrete maximum over all these smaller time interval solutions: The input feedforward support function $\rho_{u_{\mathrm{ff}}}$ for each time interval $\tau^{(i)}$ with $0 \leq i \leq m - 1$ can be directly computed (see (16) and (8)). For $\tilde{\rho}_{\Delta u}$, we can replace the approximated support function over a short time interval with the approximated support function at a finite number of time points for a large enough q.
- (20c): Analogous to input constraints (see (20b)).
- (20d): We enforce $P \in \mathcal{P}_\gamma(\bar{P}) \cap \mathcal{P}$ using the trust-region radius $\gamma \in (0, 1]$ as defined in Section IV-B to construct a trust region for the feedforward parameters, ensuring that $\tilde{\mathcal{R}}_{x_{\mathrm{ff}}}(t, P)$ is accurate and $P \in \mathcal{P} = [-1, 1]^{m n_u \times a}$.
- (20e) and (20f): Let $\mathcal{Q} \subset \mathbb{S}_{++}^{n_x \times n_x}$ and $\mathcal{R} \subset \mathbb{S}_{++}^{n_u \times n_u}$ be two bounded sets. We define two generating functions $h_Q(M) \in \mathbb{S}^{n_x \times n_x}$, where $h_Q(0) = 0$ with M being a dependent factor matrix, and $h_R(N) \in \mathbb{S}^{n_u \times n_u}$, where $h_R(0) = 0$ with N being a dependent factor matrix. Similarly to the trust-region radius γ, which limits the domain of the feedforward parameters P, we introduce the trust-region radius $\eta \in (0, 1]$ to confine $Q \in \mathcal{Q}$ and $R \in \mathcal{R}$ to smaller sets around $\bar{Q}$ and $\bar{R}$. Naturally, $\mathcal{Q}$ and h_Q as well as $\mathcal{R}$ and h_R need to be chosen such that (20e) and (20f) can be reformulated as smooth constraints in Q and R only.

To avoid the semi-definite constraints (20e) and (20f), Q and R can be chosen as diagonal matrices with positive entries as done in this article or as strictly diagonally dominant matrices with positive diagonal entries as a sufficient (but not necessary) condition for positive semi-definiteness (follows from the Gershgorin Circle Theorem [66, Th. 0]). Note that the evaluation of (20b) and (20c) requires the implicit maximization over smaller time intervals – as discussed above – to approximate the continuous maximization over $t \in [0, t_{\mathrm{f}}]$ in (22) and (23). We choose not to reformulate these maximization expressions and absolute values in (20a) with additional auxiliary variables for easier readability. With these reformulations, (20) is a non-convex, continuously differentiable optimization problem.

B. TRUST-REGION ALGORITHM

We now propose our novel trust-region algorithm: In each iteration, we solve the subproblem in (20), evaluate the newly computed controller on a subsequently defined cost function using overapproximative reachability analysis, and accept the step only if the cost decreases. At the end of each iteration, we tune γ and η so that future solutions to (20) also decrease the cost. We describe the solution procedure in Algorithm 1, where each step is explained in detail subsequently.

a) Initialization (l. 1–2): We use the approach from [50], [51] to generate an initial guess $\bar{P}$, initially set $\bar{Q} = I_{n_x}$ and $\bar{R} = I_{n_u}$ (or any other user-defined initial guess) and start with the complete range of P, Q, and R by setting $\gamma = \eta = 1$ (or any other user-defined values).

b) Initial evaluation of solution (l. 3): At a critical point $(\hat{z}, \hat{s})$ of (20), it can be shown (see proof of Theorem 1) that $\|\hat{s}\|_1 = \|\max(0, \tilde{g}(\hat{z}))\|_1$, where $\tilde{g}(z) = \begin{bmatrix} \tilde{g}_{\mathcal{U}}(z)^T, & \tilde{g}_{\mathcal{X}}(z)^T \end{bmatrix}^T$ collects (22) and (23) in a vector. Therefore, let $\tilde{J}(\hat{z}) = \tilde{J}_{\mathrm{TR}}(\hat{z}, \max(0, \tilde{g}(\hat{z})))$.

However, since $\tilde{J}(\hat{z})$ is only an approximated cost, we need to evaluate it using overapproximative reachability analysis to measure the actual cost of the controller. To preserve dependencies between the combined state, feedforward state, and feedforward input for a given initial state (similar to the

Algorithm 1: Combined Synthesis.

1: $\bar{P}$ (see [50], [51]), $\bar{Q} = I_{n_x}$, $\bar{R} = I_{n_u}$, $\gamma = 1$, $\eta = 1$
2: $\bar{z}^T = \begin{bmatrix} \bar{P}_{(:)}^T, & \bar{Q}_{(:)}^T, & \bar{R}_{(:)}^T \end{bmatrix}$
3: $[\bar{J}, \bar{\mathcal{R}}_{\text{ext}}] = \text{EVALSOL}(\bar{z})$ ▷ Section VI-B-b
4: **for** $k = 1; k \leq k_{\max}; k = k + 1$ **do**
5: $\bar{\mathcal{R}}_{x_{\text{ff}}} = \text{REACHFF}(\bar{P}, \gamma, \bar{\mathcal{R}}_{\text{ext}})$ ▷ Section IV
6: $\hat{z} = \text{TRSUBPROBLEM}(\bar{z}, \bar{\mathcal{R}}_{x_{\text{ff}}}, \bar{\mathcal{R}}_{\text{ext}}, \gamma, \eta)$ ▷ (20)
7: $[\hat{J}, \hat{\mathcal{R}}_{\text{ext}}] = \text{EVALSOL}(\hat{z})$ ▷ Section VI-B-b
8: $[\gamma, \eta] =$
9: $\text{TUNEPARAMS}(\gamma, \eta, \bar{z}, \hat{z}, \bar{\mathcal{R}}_{x_{\text{ff}}}, \bar{\mathcal{R}}_{\text{ext}}, \hat{\mathcal{R}}_{\text{ext}})$ ▷ Theorem 1
10: **if** $\hat{J} < \bar{J}$ **then**
11: $\lambda = \text{TOL}(\hat{J}, \bar{J})$ ▷ (25)
12: $\bar{J} = \hat{J}, \bar{z} = \hat{z}, \bar{\mathcal{R}}_{\text{ext}} = \hat{\mathcal{R}}_{\text{ext}}$
13: **if** $\lambda \leq \mu$ **then** ▷ (25)
14: **break**
15: **end if**
16: **end if**
17: **end for**
18: **return** $\bar{z}, \bar{\mathcal{R}}_{\text{ext}}$

reachability computation in Section IV-B), we extend the system state and compute the extended reachable set $\mathcal{R}_{\text{ext}}(t, z)$ based on the extended flow equation

$$\begin{bmatrix} \dot{x} \\ \dot{x}_{\text{ff}} \\ \dot{u}_{\text{ff}} \end{bmatrix} = \begin{bmatrix} f(x, u_{\text{ff}} + K(x - x_{\text{ff}}), w) \\ f(x_{\text{ff}}, u_{\text{ff}}, 0) \\ 0 \end{bmatrix},$$

where its zonotope overapproximation is given by

$$\mathcal{Z}_{\text{ext}}(t, z) = \left\langle \begin{bmatrix} c_x(t, z) \\ c_{x_{\text{ff}}}(t, P) \\ c_{u_{\text{ff}}}(t, P) \end{bmatrix}, \begin{bmatrix} G_x(t, z) \\ G_{x_{\text{ff}}}(t, P) \\ G_{u_{\text{ff}}}(t, P) \end{bmatrix} \right\rangle_Z. \tag{24}$$

With (24), we can define the actual control cost $J(z)$, which is given analogously to $\tilde{J}(z)$ but where sets (and corresponding support functions) are replaced by their formally correct versions contained in (24).

c) Feedforward reachable set computation (l. 5): If the maximum number of iterations $k_{\max} \in \mathbb{N}_+$ is not yet reached, we compute the parameterized, undisturbed feedforward reachable set for $P \in \mathcal{P}_\gamma(\bar{P})$, as described in Section IV, at the start of each iteration.

d) Computation & evaluation of new candidate controller (l. 6–7): As indicated in Sections IV and V, $\mathcal{R}_{\text{ext}}(t, \bar{z})$ and its zonotope overapproximation $\mathcal{Z}_{\text{ext}}(t, \bar{z})$ are required in (20) to construct the feedforward approximation in (11) and the support function approximation in (19), respectively. With $\hat{z}$ being a solution to (20), we then compute its cost $J(\hat{z})$, which is used subsequently to determine if the current step is accepted. If the extended, overapproximative reachable set for the computation of $J(\hat{z})$ cannot be computed (see, e.g., Fig. 4), we shrink both γ and η and restart the iteration (not shown in Algorithm 1).

e) Parameter tuning (l. 9): Theorem 1 discusses the tuning of the trust-region radii γ and η such that at a critical point $(\hat{z}, \hat{s})$ of (20), the objective value $\tilde{J}_{\text{TR}}(\hat{z}, \hat{s})$ approaches $J(\hat{z})$.

f) Step acceptance/rejection (l. 10–16): If $J(\hat{z}) < J(\bar{z})$ for the newly computed control parameters $\hat{z}$, the step is accepted. Furthermore, we terminate the algorithm if either the absolute or relative difference in the controller cost between two accepted steps is small enough, i.e.

$$\min\left(J(\bar{z}) - J(\hat{z}), \frac{J(\bar{z}) - J(\hat{z})}{\min(J(\bar{z}), J(\hat{z}))}\right) \leq \mu, \tag{25}$$

with user-defined tolerance $\mu \in \mathbb{R}_+$.

C. ACCURACY OF TRUST-REGION APPROXIMATION

Ideally, we want to minimize the controller cost $J(z)$. However, to avoid recomputation of the overapproximative reachable set as much as possible, we instead solve the approximated problem in (20). Thus, it is crucial that the approximated objective value $\tilde{J}_{\text{TR}}(\hat{z}, \hat{s})$ at a critical point $(\hat{z}, \hat{s})$ of the trust-region problem (20) can approximate $J(\hat{z})$ "well enough". In this section, we show that the trust-region radii γ and η can indeed be tuned such that $\tilde{J}_{\text{TR}}(\hat{z}, \hat{s})$ approximates $J(\hat{z})$ arbitrarily closely; otherwise, the approximated trust-region problem (20) potentially does not model the real cost of the controller and thus optimizing over it becomes meaningless.

For the derivation of this result, we define $\tilde{J}_{\Delta x}(z)$, which is given analogously to $\tilde{J}(z)$ but where we replace the approximated feedforward reachable set $\tilde{\mathcal{R}}_{x_{\text{ff}}}(t, P)$ and its approximated support function $\tilde{\rho}_{x_{\text{ff}}}(l, t, P)$ with their overapproximative versions from (24). Intuitively, we remove any error caused by approximating the feedforward reachable set and thus $\tilde{J}_{\Delta x}(z)$ is only inaccurate due to the approximated support functions $\tilde{\rho}_{\Delta x}$ and $\tilde{\rho}_{\Delta u}$. We are now ready to state the main result of this section.

Theorem 1 (Accurate Trust-Region Subproblem): If we adapt γ and η independently according to

$$\gamma \leftarrow \min\left(1, v\left(\max\left(e_{\text{ff},\gamma}, e_{\Delta,\gamma}\right)\right)\gamma\right), \tag{26}$$

$$\eta \leftarrow \min\left(1, v\left(e_{\Delta,\eta}\right)\eta\right), \tag{27}$$

with

$$e_{\text{ff},\gamma} = \left|\tilde{J}(\hat{z}) - \tilde{J}_{\Delta x}(\hat{z})\right|, \tag{28}$$

$$e_{\Delta,\gamma} = \left|\tilde{J}_{\Delta x}(\hat{P}, \bar{Q}, \bar{R}) - J(\hat{P}, \bar{Q}, \bar{R})\right|, \tag{29}$$

$$e_{\Delta,\eta} = \Big|\tilde{J}_{\Delta x}(\hat{z}) - \tilde{J}_{\Delta x}(\hat{P}, \bar{Q}, \bar{R}) - \left(J(\hat{z}) - J(\hat{P}, \bar{Q}, \bar{R})\right)\Big|, \tag{30}$$

where $v : [0, \infty) \mapsto \mathbb{R}_+$ is an arbitrary, monotonically decreasing function with $v(0) = \bar{c}$, $v(\underline{\psi}) = 1$, and $\forall r \geq \bar{\psi} : v(r) = \underline{c}$, where $\frac{1}{\underline{c}} > \bar{c} > 1$ and $0 \leq \underline{\psi} < \bar{\psi}$, then

$$\left|\tilde{J}_{\text{TR}}(\hat{z}, \hat{s}) - J(\hat{z})\right| \leq \epsilon,$$

is achieved after a finite number of iterations for a critical point $(\hat{z}, \hat{s})$ of (20) for $\epsilon > 0$.

Proof: We show the claim by first proving that $\left|\tilde{J}(\hat{z}) - J(\hat{z})\right| \leq \epsilon$ after following (26) and (27) for a finite number of iterations and then verifying that $\tilde{J}_{\mathrm{TR}}(\hat{z}, \hat{s}) = \tilde{J}(\hat{z})$.

It holds that

$$\begin{aligned}
&\left|\tilde{J}(\hat{z}) - J(\hat{z})\right| \\
&= \Big|\tilde{J}(\hat{z}) - \tilde{J}_{\Delta x}(\hat{z}) + \tilde{J}_{\Delta x}\left(\hat{P}, \bar{Q}, \bar{R}\right) - J\left(\hat{P}, \bar{Q}, \bar{R}\right) \\
&\quad + \tilde{J}_{\Delta x}(\hat{z}) - \tilde{J}_{\Delta x}\left(\hat{P}, \bar{Q}, \bar{R}\right) - \left(J(\hat{z}) - J\left(\hat{P}, \bar{Q}, \bar{R}\right)\right)\Big| \\
&\leq e_{\mathrm{ff},\gamma} + e_{\Delta,\gamma} + e_{\Delta,\eta}, \qquad (31)
\end{aligned}$$

due to the triangle inequality and (28) to (30). From the definition of $\tilde{J}_{\Delta x}(\hat{z})$ follows that $\lim_{\gamma \to 0} e_{\mathrm{ff},\gamma} = 0$ since $\tilde{J}_{\Delta x}(\hat{z})$ is constructed using overapproximative feedforward sets and $\lim_{\gamma \to 0} \tilde{\mathcal{R}}_{x_{\mathrm{ff}}}\left(t, \hat{P}\right) = \mathcal{R}_{x_{\mathrm{ff}}}\left(t, \bar{P}\right)$ (see (11)). Similarly, $\lim_{\gamma \to 0} e_{\Delta,\gamma} = 0$ holds since $\lim_{\gamma \to 0} \hat{P} = \bar{P}$ and because $\tilde{J}_{\Delta x}(\bar{z}) = J(\bar{z})$ by construction. Further, $\lim_{\eta \to 0} e_{\Delta,\eta} = 0$ follows from $\lim_{\eta \to 0} \tilde{J}_{\Delta x}(\hat{z}) = \tilde{J}_{\Delta x}\left(\hat{P}, \bar{Q}, \bar{R}\right)$ and $\lim_{\eta \to 0} J(\hat{z}) = J\left(\hat{P}, \bar{Q}, \bar{R}\right)$. By following (26) and (27), there is therefore always a finite number of steps until $e_{\mathrm{ff},\gamma} \leq \delta$, $e_{\Delta,\gamma} \leq \delta$, and $e_{\Delta,\eta} \leq \delta$, with $\delta > 0$, from which $\left|\tilde{J}(\hat{z}) - J(\hat{z})\right| \leq \epsilon$ follows from (31) and because δ is arbitrary.

Lastly, (20) is always regular at a critical point $(\hat{z}, \hat{s})$ (proof omitted due to space considerations) and thus any critical point necessarily fulfills the Karush-Kuhn-Tucker (KKT) conditions. The relevant KKT conditions of (20) in s for this proof are

$$\sigma 1_o - \hat{\mu} - \hat{\lambda} = 0, \qquad (32)$$

$$\hat{\mu}_k \left(\tilde{g}_k(\hat{z}) - \hat{s}_k\right) = 0, \ 1 \leq k \leq o, \qquad (33)$$

$$\hat{\lambda}_k \left(-\hat{s}_k\right) = 0, \ 0 \leq k \leq o, \qquad (34)$$

where $\hat{\mu} \in \mathbb{R}^o_{\geq 0}$ and $\hat{\lambda} \in \mathbb{R}^o_{\geq 0}$ are the constraint multipliers at the critical point of the collected constraints in (20b), (20c) and (20g), respectively, and where $\tilde{g}(\hat{z}) \in \mathbb{R}^o$. Let $\mathcal{I} = \{i \in \{1, \ldots, o\} \mid \tilde{g}_i(\hat{z}) \leq 0\}$ and $\mathcal{J} = \{j \in \{1, \ldots, o\} \mid \tilde{g}_j(\hat{z}) > 0\}$. Since $\hat{\mu}_k = 0$ if $\tilde{g}_k(\hat{z}) < 0$ due to (33) and (20g) for $1 \leq k \leq o$, it follows that $\forall i \in \mathcal{I} : \hat{\mu}_i \tilde{g}_i(\hat{z}) = 0$. Further, it holds that $\forall j \in \mathcal{J} : \tilde{g}_j(\hat{z}) > 0 \overset{(20b),(20c)}{\Rightarrow} \hat{s}_j > 0 \overset{(34)}{\Rightarrow} \hat{\lambda}_j = 0 \overset{(32)}{\Rightarrow} \hat{\mu}_j = \sigma$. Thus, it follows from (33) that

$$\begin{aligned}
\hat{\mu}^T \tilde{g}(\hat{z}) &= \sum_{i \in \mathcal{I}} \hat{\mu}_i \tilde{g}_i(\hat{z}) + \sum_{j \in \mathcal{J}} \hat{\mu}_j \tilde{g}_j(\hat{z}) = \sum_{j \in \mathcal{J}} \sigma \left|\tilde{g}_j(\hat{z})\right| \\
&= \sigma \left\|\max\left(0, \tilde{g}(\hat{z})\right)\right\|_1 = \sigma 1_o^T \hat{s},
\end{aligned}$$

and therefore $\tilde{J}(\hat{z}) = \tilde{J}_{\mathrm{TR}}(\hat{z}, \hat{s})$, concluding the proof. ■

Tuning according to Theorem 1 requires one additional overapproximative reachable set computation at $P = \hat{P}$, $Q = \bar{Q}$, and $R = \bar{R}$. Alternatively, one can set $e_{\Delta,\gamma} = e_{\Delta,\eta} = e_\Delta$ with $e_\Delta = \left|\tilde{J}_{\Delta x}(\hat{z}) - J(\hat{z})\right|$ to avoid that computation at the cost of possibly unnecessarily shrinking either γ or η (since both are shrunk equally).

D. COMPUTATIONAL COMPLEXITY

Since the maximum number of iterations in Algorithm 1 is fixed, we only consider the computational complexity in the state dimension for one iteration of Algorithm 1 in this section. As Section VII (see Tables 1 and 2) indicates, the proposed algorithm typically terminates within a small number of iterations; thus, limiting the number of iterations to a reasonably small number often does not impede performance in practice. In this section, we show that the complexity of one iteration in Algorithm 1 is at most $O\left(n_x^5 + \upsilon(\epsilon) m \left(n^6 + n^4 n_z^2 + q l \xi n^\omega n_z^2\right)\right)$ where $n = \max(n_x, n_u)$ and $O(n^\omega)$ with $\omega \geq 2$ is the complexity of multiplying two $n \times n$ matrices.

a) Reachable set computations: The computation of reachable sets (l. 5 for the approximated feedforward reachable set; l. 3 and 7 for the extended overapproximative reachable set) has complexity $O\left(n_x^5\right)$ [67, Sec. 4.1.4].

b) Trust-region subproblem: The total number of function evaluations $\upsilon(\epsilon)$ with $\upsilon : \mathbb{R}_+ \mapsto \mathbb{N}_+$ required to solve (20) is polynomially dependent on the inverse of the requested solution accuracy $\epsilon > 0$ if second-order methods are used [68]. Thus, we continue to derive the computational complexity for one objective and constraint function evaluation of (20), which is dominated by the computation of the Hessian matrix for each element of the approximated disturbance tube shape matrix $Q_{\Delta x}(t, z)$ using (18). For its evaluation, we compute the Hessian matrix of each element of $K^{(i)}$ for $0 \leq i \leq m - 1$ in terms of $z \in \mathbb{R}^{n_z}$, where the complexity of one evaluation is $O\left(n^6 + n^4 n_z^2\right)$; we omit the proof to keep the presentation compact. Further, we need to evaluate (18) mq times, where one evaluation is dominated by the computation of the Hessian matrix of each element of the integral, within which the Hessian matrix computation for each matrix element of $e^{\bar{A}(t,z)\delta}$ dominates with complexity $O\left(\xi n^\omega n_z^2\right)$. Here, $\xi \in \mathbb{N}_+$ denotes the finite number of terms from the infinite series of $e^{\bar{A}(t,z)\delta}$ to compute the matrix exponential accurately enough and $O\left(n^\omega n_z^2\right)$ is the complexity for each of the ξ terms, which follows from the complexity of multiplying two matrices but where each element multiplication is the outer product of two n_z-dimensional vectors. We compute the integral by solving the corresponding ordinary differential equation (ODE) with complexity $O\left(l \xi n^\omega n_z^2\right)$, where $l \in \mathbb{N}_+$ collects the fixed number of function evaluations per ODE step (e.g., four evaluations for the classical Runge-Kutta method) and the total number of ODE steps required, which can be assumed to be fixed if a solver with a fixed step size is used. Thus, evaluating (18) mq times over $\upsilon(\epsilon)$ optimization iterations yields the complexity $O\left(\upsilon(\epsilon) m \left(n^6 + n^4 n_z^2 + q l \xi n^\omega n_z^2\right)\right)$.

VII. EXPERIMENTS

We demonstrate the applicability of the novel iterative polynomial reachset optimal control (iPROC) approach by comparing its performance against the reachset optimal control (ROC) algorithm from [51] (both implemented in MATLAB) and the robustly complete control synthesis (ROCS)

C++ toolbox[1] [69], [70]. We first present a water tank benchmark [71] in more detail since it is easily extendable to an arbitrary number of states to compare scalability. In Section VII-B, we then shortly introduce other benchmarks from previous work [51] and from the applied verification for continuous and hybrid systems (ARCH)[2] competition.

We run all experiments on an Intel(R) Core(TM) i7-8650U processor with 24 GB of RAM. We note that ROC uses parallel computing to numerically approximate gradients and Hessian matrices while iPROC computes them analytically and thus does not use parallel computing. We use the interior point optimizer (IPOPT) [72], specifically the MATLAB interface included in the OPTI toolbox[3], for iPROC and MATLAB's `fmincon` for ROC. Throughout this section, we assume $\sigma = 1000$, use a linear feedforward controller with $E = [0, \quad I_{n_x}]$, and set $\mu = 1 \times 10^{-2}$. Overapproximative reachable sets are computed using the continuous reachability analyzer (CORA) toolbox[4] [73]. CORA does not consider floating point errors for the reachable set computation. To account for these, one could integrate interval arithmetic as done in [74]. Our proposed algorithm will be implemented in the next release of the automated reachset optimal control (AROC) toolbox[5] [75].

A. WATER TANKS

A system of n water tanks is given by (see [71] for details)

$$\dot{x}_1 = u + w - k\sqrt{2gx_1},$$
$$\dot{x}_i = k\left(\sqrt{2gx_{i-1}} - \sqrt{2gx_i}\right),$$

with $2 \leq i \leq n$ and where $x_k = \frac{h_k}{\mathrm{m}} \in \mathbb{R}_{\geq 0}$ for $1 \leq k \leq n$, h_k is the water level of the k-th tank, $u = \frac{l}{\mathrm{m}^3/\mathrm{s}} \in \mathbb{R}_{\geq 0}$ and l is the inflow into the first tank, $w = \frac{\nu}{\mathrm{m}^3/\mathrm{s}} \in \mathbb{R}$ and ν is the uncontrollable inflow into the first tank, and $k = 0.015$ and $g = 9.81$.

We set $m = 2, n = 2, \mathcal{X}^{(0)} = \left\{\bar{x}^{(0)} \;\middle|\; \left\|\bar{x}^{(0)} - x^{(0)}\right\|_\infty \leq 1\right\}$, $x^{(0)} = x_\mathrm{f} = 10 \cdot 1_n, \mathcal{U} = \langle 1, 1\rangle_Z, \mathcal{W} = \langle 0, 0.02\rangle_Z, t_\mathrm{f} = 120\,\mathrm{s}$, and the final state constraint is $\mathcal{X}_\mathrm{f} = \mathcal{X}^{(0)}$. Fig. 3 compares the performance of iPROC against ROC and ROCS, where all approaches find feasible solutions for the initial set (also see Table 2). Since the ROCS approach does not directly realize reachable set minimization at a given final time to the best of our knowledge, we manually tried to find the smallest goal region for which a controller for all initial states, here $G_\mathrm{ROCS} = \left\langle [10, \quad 10]^T, \mathrm{diag}([0.03, \quad 0.03])\right\rangle_Z$, exists. Both ROC and ROCS fail to find a feasible solution if algorithm parameters are set incorrectly.

Furthermore, Table 1 displays the computation times and sizes of the final reachable set $\mathcal{R}_x(t_\mathrm{f}) = \langle c(t_\mathrm{f}), G(t_\mathrm{f})\rangle_Z$ (where

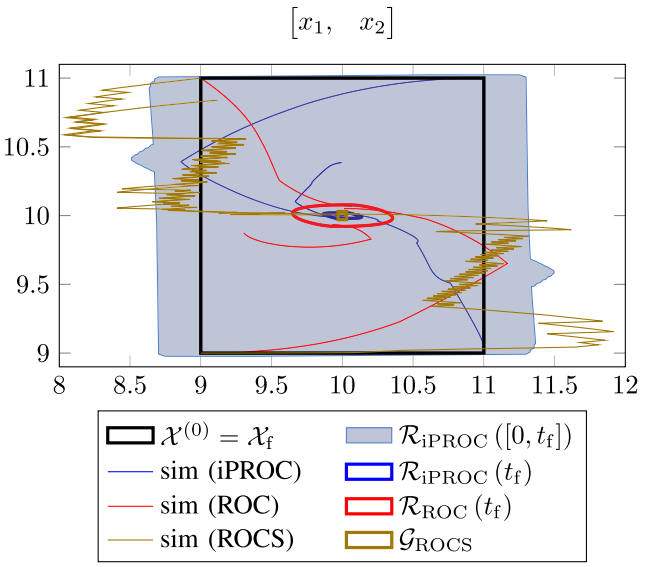

FIGURE 3. Comparison of our novel iPROC approach with ROC and ROCS for the tank benchmark. The best achievable goal region for ROCS is denoted by $\mathcal{G}_\mathrm{ROCS}$.

TABLE 1. Scalability comparison of iPROC, ROC, and ROCS for the tank example using 2 to 8 tanks with $m = 2$ and no final state constraints.

nr. of tanks	2	4	6	8
time [s] (iPROC)	142	218	499	624
time [s] (ROC)	98	383	2141	5596
time [s] (ROCS)	23	–	–	–
csize $(\mathcal{R}_x(t_\mathrm{f}))$ (iPROC)	0.18	1.66	3.76	5.92
csize $(\mathcal{R}_x(t_\mathrm{f}))$ (ROC)	0.49	2.01	4.01	6.09
nr. of iter. (iPROC)	13	8	8	8

We terminate all computations taking longer than 10800 s = 3 h and denote the timeout with "–". The goal region of ROCS is $G_\mathrm{ROCS} = \left\langle [10, \quad 10]^T, \mathrm{diag}([0.03, \quad 0.03])\right\rangle_Z$.

appropriate), defined by

$$\mathrm{csize}(\mathcal{R}_x(t_\mathrm{f})) = \left\| \left[c(t_\mathrm{f}) - x_\mathrm{f}, \quad G(t_\mathrm{f})\right]_{(:)} \right\|_1,$$

for up to 8 tanks. We omit final state constraints to avoid infeasibility for higher dimensions so that only input constraints are active. Our novel approach scales better with an increasing number of dimensions compared to both ROC and ROCS while mostly producing smaller final reachable sets. Further, Table 1 also displays the number of trust-region iterations of Algorithm 1 of the novel iPROC approach for the tank example, which indicates that the number of iterations does not really grow with the number of state variables.

Additionally, Fig. 4 shows the solve progress of iPROC, which demonstrates the accurate approximation of $J(z)$ by $\tilde{J}(z)$ as derived in Theorem 1.

B. BENCHMARKS FROM PREVIOUS WORK AND THE ARCH COMPETITION

In this section, we first look at the car benchmark from [51], where a left turn maneuver using a kinematic model is to be performed under input and final state constraints. We also

[1][Online]. Available: https://git.uwaterloo.ca/hybrid-systems-lab/rocs/-/tree/master/
[2][Online]. Available: https://cps-vo.org/group/ARCH
[3][Online]. Available: https://github.com/jonathancurrie/OPTI
[4][Online]. Available: https://cora.in.tum.de/
[5][Online]. Available: https://aroc.in.tum.de/

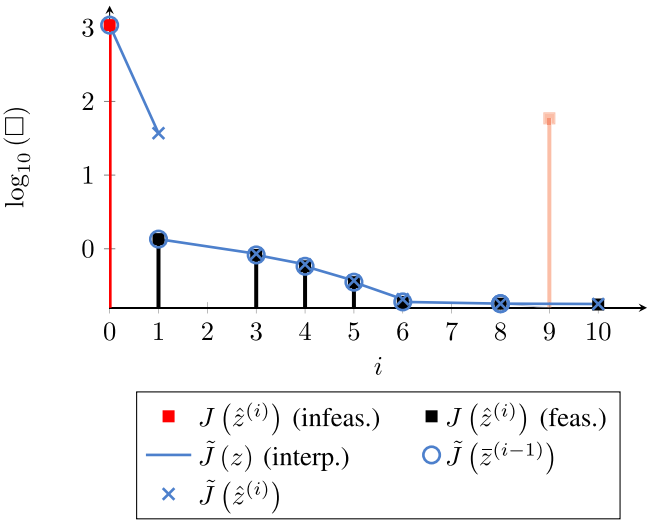

FIGURE 4. Visualization of the solver progress and the approximation quality of the trust-region subproblem in (20) for the tank benchmark. Values between $\tilde{J}(\bar{z}^{(i-1)})$ and $\tilde{J}(\hat{z}^{(i)})$, $1 \leq i \leq 10$ – where $\bar{z}^{(i-1)}$ and $\hat{z}^{(i)}$ are the initial and critical points of the subproblem in (20) – are linearly interpolated ($\hat{z}^{(0)} = \bar{z}^{(0)}$). For steps $i = 2$ and $i = 7$, there were negative water values and thus no controller cost could be computed. The opaque value at $i = 9$ denotes a rejected iteration.

TABLE 2. Collection of benchmarks.

	tank	car	space
time [s] **(iPROC)**	137	1316	1324
time [s] **(ROC)**	43	776	989
csize $(\mathcal{R}_x(t_f))$ **(iPROC)**	0.18	0.14	0.15
csize $(\mathcal{R}_x(t_f))$ **(ROC)**	0.43	0.39	0.21
nr. of iter. (iPROC)	10	9	8

synthesize controllers for a spacecraft approach from [76, Sec. 3.6], where a spacecraft attempts docking with another spacecraft under input, state, and final state constraints. To keep the presentation compact, we only show computation times and the sizes of the final, closed-loop reachable sets in Table 2 for each benchmark. We were not able to find a controller steering all states from the initial set into a reasonably small target state for any of the following benchmarks using ROCS in less than 3 h and thus ROCS is omitted.

a) Kinematic car: We compute a left turn for the kinematic car benchmark from [51], which includes input and final state constraints. While both iPROC and ROC were able to find a feasible solution, iPROC was able to find a smaller final reachable set without manual tuning (see Table 2).

b) Space rendezvous: In this benchmark, a controller for the rendezvous attempt of two spacecraft as described in [76, Sec. 3.6] is synthesized, where the first and last two states describe the position and velocity of the controlled spacecraft in a common orbital plane, respectively. This benchmark includes input, state, and final state constraints, where the state constraints consist of the approaching spacecraft being required to stay within a cone defining the line of sight and respecting velocity constraints. We choose $m = 5$, $t_f = 200$ s, an initial set $\mathcal{X}^{(0)} = \langle x^{(0)}, \text{diag}([2, \quad 2, \quad 0.1, \quad 0.1]) \rangle_Z$ with $x^{(0)} = [-95, \quad -30, \quad 0, \quad 0]^T$, target state $x_f = [-1, \quad 0, \quad 0, \quad 0]^T$, $\mathcal{X}_f = \mathcal{X}^{(0)} \oplus \{-x^{(0)} + x_f\}$, and $\mathcal{W} = 0.05\,\mathcal{U}$, where $\mathcal{U}$ and $\mathcal{X}$ are given as in [76, Sec. 3.6], but we replace the velocity constraints in $\mathcal{X}$ with their parallelotope underapproximation for computational reasons. While both iPROC and ROC found feasible solutions, iPROC again achieves a smaller final reachable set without the need for manual tuning (see Table 2).

VIII. CONCLUSION

We introduce a novel, formally verified, polynomial control synthesis approach for disturbed nonlinear systems that simultaneously synthesizes a piecewise constant feedforward controller and a continuous-time state feedback controller. In contrast to existing work, we avoid the introduction of algorithm parameters which require expert knowledge to tune. We achieve this for the first time by combining the synthesis for the feedforward and feedback controller into a single optimization problem and using a trust-region approach to iteratively ensure the accuracy of this optimization problem. Additionally, we show that this optimization problem can approximate the formally correct controller cost arbitrarily closely, and furthermore prove the polynomial complexity of our novel synthesis approach in the state dimension for each trust-region iteration. Numerical examples indicate that our novel approach achieves similar performance – or even outperforms previous work – while not requiring manual tuning of algorithm parameters, making it more easily applicable by non-experts.

APPENDIX

For matrices $A \in \mathbb{R}^{o_1 \times o_2}$ and $B \in \mathbb{R}^{c_1 \times c_2}$, the Kronecker product is denoted by $A \boxtimes B \in \mathbb{R}^{o_1 c_1 \times o_2 c_2}$. Further, the Kronecker sum for two square matrices $C \in \mathbb{R}^{c \times c}$ and $D \in \mathbb{R}^{d \times d}$ is defined as $C \boxplus D = C \boxtimes I_d + I_c \boxtimes D$.

For a controllable LTI system with system matrix $A(P) \in \mathbb{R}^{n_x \times n_x}$ and input matrix $B(P) \in \mathbb{R}^{n_x \times n_u}$, both parameterized in the feedforward parameters P, the optimal gain matrix is

$$K(z) = -R^{-1} B(P)^T X(z), \tag{35}$$

where $X(z) \in \mathbb{S}^{n_x \times n_x}_{++}$ is the positive definite solution to the Riccati equation

$$\begin{aligned} F(z) = {} & A(P)^T X(z) + X(z) A(P) \\ & - X(z) B(P) R^{-1} B(P)^T X(z) + Q \\ = {} & 0, \end{aligned} \tag{36}$$

and where $z = \left[P_{(:)}^T, \quad Q_{(:)}^T, \quad R_{(:)}^T\right]^T$. We are now ready to state the main result of the appendix.

Theorem 2 (Continuous differentiability): Let an n_x-dimensional, controllable LTI system with system matrix $A(P)$ and input matrix $B(P)$, both k-times differentiable in $P \in [-1, 1]^{mn_u \times a}$ for each matrix element, be given. Further, denote with K and X the corresponding solution to (35) and (36) using $Q \in \mathbb{S}_{++}^{n_x \times n_x}$ and $R \in \mathbb{S}_{++}^{n_u \times n_u}$, respectively. The gain matrix K is then k-times differentiable with respect to $z = \begin{bmatrix} P_{(:)}^T, & Q_{(:)}^T, & R_{(:)}^T \end{bmatrix}^T$.

Proof: For readability, we omit the arguments of functions where convenient.

We first prove that $X_{(:)}(z)$ exists, is unique, and is k-times differentiable with respect to z, which follows from the implicit function theorem if $\frac{dF_{(:)}(z)}{dX_{(:)}(z)}$ is invertible for all z. The first differential of F from (36) with respect to X is

$$dF = dX\,(A + BK) + (A + BK)^T\,dX,$$

which, after vectorization, yields $\frac{dF_{(:)}}{dX_{(:)}} = A_{\mathrm{cl}}^T \boxplus A_{\mathrm{cl}}^T$ [77, Th. 18.1], where $A_{\mathrm{cl}} = A + BK$. Since A_{cl} only has eigenvalues with negative real part by design, all n_x^2 eigenvalues of $A_{\mathrm{cl}}^T \boxplus A_{\mathrm{cl}}^T$ also have negative real parts [78, Th. 13.16] and thus $(A_{\mathrm{cl}}^T \boxplus A_{\mathrm{cl}}^T)^{-1}$ exists for all P.

The k-times differentiability of K then follows directly from its definition in (35) since X is k-times differentiable and $R \in \mathbb{S}_{++}^{n_u \times n_u}$. ■

REFERENCES

[1] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi reachability: A brief overview and recent advances," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 2242–2253.

[2] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *Proc. Conf. Hybrid Syst.: Comput. Control*, 2015, pp. 11–20.

[3] D. Kalise, S. Kundu, and K. Kunisch, "Robust feedback control of nonlinear PDEs by numerical approximation of high-dimensional Hamilton–Jacobi–Isaacs equations," *SIAM J. Appl. Dynamical Syst.*, vol. 19, no. 2, pp. 1496–1524, 2020.

[4] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. Berlin, Germany: Springer, 2009.

[5] G. Pola, A. Girard, and P. Tabuada, "Approximately bisimilar symbolic models for nonlinear control systems," *Automatica*, vol. 44, no. 10, pp. 2508–2516, 2008.

[6] G. Pola and P. Tabuada, "Symbolic models for nonlinear control systems: Alternating approximate bisimulations," *SIAM J. Control Optim.*, vol. 48, no. 2, pp. 719–733, 2009.

[7] G. Reissig, "Computing abstractions of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 56, no. 11, pp. 2583–2598, 2011.

[8] G. Reissig and M. Rungger, "Symbolic optimal control," *IEEE Trans. Autom. Control*, vol. 64, no. 6, pp. 2224–2239, 2019.

[9] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Trans. Autom. Control*, vol. 53, no. 1, pp. 287–297, 2008.

[10] A. Girard, "Controller synthesis for safety and reachability via approximate bisimulation," *Automatica*, vol. 48, no. 5, pp. 947–953, 2012.

[11] A. Girard and S. Martin, "Motion planning for nonlinear systems using hybridizations and robust controllers on simplices," in *Proc. IEEE Conf. Decis. Control*, 2008, pp. 239–244.

[12] Y. Bai and K. Mallik, "Accurate abstractions for controller synthesis with non-uniform disturbances," in *Proc. Conf. Formal Eng. Methods*, 2020, pp. 297–307.

[13] L. C. G. J. M. Habets, P. J. Collins, and J. H. V. Schuppen, "Reachability and control synthesis for piecewise-affine hybrid systems on simplices," *IEEE Trans. Autom. Control*, vol. 51, no. 6, pp. 938–948, 2006.

[14] C. Belta and L. C. G. J. M. Habets, "Controlling a class of nonlinear systems on rectangles," *IEEE Trans. Autom. Control*, vol. 51, no. 11, pp. 1749–1759, 2006.

[15] D. Förstner, M. Jung, and J. Lunze, "A discrete-event model of asynchronous quantised systems," *Automatica*, vol. 38, no. 8, pp. 1277–1286, 2002.

[16] A. Bicchi, A. Marigo, and B. Piccoli, "On the reachability of quantized control systems," *IEEE Trans. Autom. Control*, vol. 47, no. 4, pp. 546–563, 2002.

[17] A. Girard, G. Pola, and P. Tabuada, "Approximately bisimilar symbolic models for incrementally stable switched systems," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 116–126, 2010.

[18] G. Pola, P. Pepe, M. D. D. Benedetto, and P. Tabuada, "Symbolic models for nonlinear time-delay systems using approximate bisimulations," *Syst. Control Lett.*, vol. 59, no. 6, pp. 365–373, 2010.

[19] M. Zamani, G. Pola, M. Mazo, and P. Tabuada, "Symbolic models for nonlinear control systems without stability assumptions," *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1804–1809, 2012.

[20] J. A. DeCastro and H. Kress-Gazit, "Synthesis of nonlinear continuous controllers for verifiably correct high-level, reactive behaviors," *J. Robot. Res.*, vol. 34, no. 3, pp. 378–394, 2015.

[21] J. Liu and N. Ozay, "Finite abstractions with robustness margins for temporal logic-based control synthesis," *Nonlinear Anal.: Hybrid Syst.*, vol. 22, pp. 1–15, 2016.

[22] P. Nilsson and N. Ozay, "Incremental synthesis of switching protocols via abstraction refinement," in *Proc. IEEE Conf. Decis. Control*, 2014, pp. 6246–6253.

[23] G. Reissig, A. Weber, and M. Rungger, "Feedback refinement relations for the synthesis of symbolic controllers," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1781–1796, 2017.

[24] R. Majumdar, N. Ozay, and A. Schmuck, "On abstraction-based controller design with output feedback," in *Proc. Conf. Hybrid Syst.: Comput. Control*, 2020, pp. 1–11.

[25] E. M. Wolff and R. M. Murray, "Optimal control of nonlinear systems with temporal logic specifications," in *Proc. Int. Symp. Robot. Res.*, 2016, pp. 21–37.

[26] I. Papusha, J. Fu, U. Topcu, and R. M. Murray, "Automata theory meets approximate dynamic programming: Optimal control with temporal logic constraints," in *Proc. IEEE Conf. Decis. Control*, 2016, pp. 434–440.

[27] T. Moor and J. Raisch, "Abstraction based supervisory controller synthesis for high order monotone continuous systems," in *Modelling, Analysis, and Design of Hybrid Systems*. Berlin, Germany: Springer, 2002, pp. 247–265.

[28] E. Feron, P. Apkarian, and P. Gahinet, "Analysis and synthesis of robust control systems via parameter-dependent Lyapunov functions," *IEEE Trans. Autom. Control*, vol. 41, no. 7, pp. 1041–1046, 1996.

[29] A. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. Eur. Control Conf.*, 2019, pp. 3420–3431.

[30] F. S. Barbosa, L. Lindemann, D. V. Dimarogonas, and J. Tumova, "Provably safe control of Lagrangian systems in obstacle-scattered environments," in *Proc. IEEE Conf. Decis. Control*, 2020, pp. 2056–2061.

[31] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. IEEE Conf. Decis. Control*, 2014, pp. 6271–6278.

[32] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier Lyapunov functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, no. 4, pp. 918–927, 2009.

[33] M. Z. Romdlony and B. Jayawardhana, "Stabilization with guaranteed safety using control Lyapunov–barrier function," *Automatica*, vol. 66, pp. 39–47, 2016.

[34] R. Tedrake, "LQR-trees: Feedback motion planning on sparse randomized trees," in *Proc. Robot.: Sci. Syst.*, 2009.[Online]. Available: https://roboticsproceedings.org/rss05/p3.html

[35] P. Reist and R. Tedrake, "Simulation-based LQR-trees with input and state constraints," in *Proc. IEEE Conf. Robot. Automat.*, 2010, pp. 5504–5510.

[36] S. Sadraddini and R. Tedrake, "Sampling-based polytopic trees for approximate optimal control of piecewise affine systems," in *Proc. IEEE Conf. Robot. Automat.*, 2019, pp. 7690–7696.

[37] M. Morari and J. H. Lee, "Model predictive control: Past, present and future," *Comput. Chem. Eng.*, vol. 23, no. 4-5, pp. 667–682, 1999.

[38] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, WI, USA: Nob Hill Publishing, 2009.
[39] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi, "Tube-based robust nonlinear model predictive control," *J. Robust Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.
[40] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, "Robust tube-based MPC for tracking of constrained linear systems with additive disturbances," *J. Process Control*, vol. 20, no. 3, pp. 248–260, 2010.
[41] D. Q. Mayne, S. V. Raković, R. Findeisen, and F. Allgöwer, "Robust output feedback model predictive control of constrained linear systems: Time varying case," *Automatica*, vol. 45, no. 9, pp. 2082–2087, 2009.
[42] S. V. Rakovic, B. Kouvaritakis, M. Cannon, C. Panos, and R. Findeisen, "Parameterized tube model predictive control," *IEEE Trans. Autom. Control*, vol. 57, no. 11, pp. 2746–2761, 2012.
[43] D. Q. Mayne, S. V. Raković, R. Findeisen, and F. Allgöwer, "Robust output feedback model predictive control of constrained linear systems," *Automatica*, vol. 42, no. 7, pp. 1217–1222, 2006.
[44] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," in *Proc. Nonlinear Model Predictive Control*, 2009, pp. 345–369.
[45] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.
[46] F. Gruber and M. Althoff, "Scalable robust model predictive control for linear sampled-data systems," in *Proc. IEEE Conf. Decis. Control*, 2019, pp. 438–444.
[47] M. N. Zeilinger, D. M. Raimondo, A. Domahidi, M. Morari, and C. N. Jones, "On real-time robust model predictive control," *Automatica*, vol. 50, no. 3, pp. 683–694, 2014.
[48] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Technische Universität München, Munich, Germany, 2010.
[49] B. Schürmann and M. Althoff, "Optimal control of sets of solutions to formally guarantee constraints of disturbed linear systems," in *Proc. Amer. Control Conf.*, 2017, pp. 2522–2529.
[50] B. Schürmann and M. Althoff, "Guaranteeing constraints of disturbed nonlinear systems using set-based optimal control in generator space," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11515–11522, 2017.
[51] B. Schürmann and M. Althoff, "Optimizing sets of solutions for controlling constrained nonlinear systems," *IEEE Trans. Autom. Control*, vol. 66, no. 3, pp. 981–994, 2021.
[52] V. Gaßmann and M. Althoff, "Verified polynomial controller synthesis for disturbed nonlinear systems," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 85–90, 2021.
[53] A. Platzer and E. Clarke, "The image computation problem in hybrid systems model checking," in *Hybrid Systems: Computation and Control*. Berlin, Germany: Springer, 2007, pp. 473–486.
[54] M. Althoff, "Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets," in *Proc. Conf. Hybrid Syst.: Comput. Control*, 2013, pp. 173–182.
[55] N. Kochdumper and M. Althoff, "Sparse polynomial zonotopes: A novel set representation for reachability analysis," *IEEE Trans. Autom. Control*, vol. 66, no. 9, pp. 4043–4058, 2021.
[56] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal toolbox (ET)," in *Proc. IEEE Conf. Decis. Control*, 2006, pp. 1498–1503.
[57] A. Kopetzki, B. Schürmann, and M. Althoff, "Methods for order reduction of zonotopes," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 5626–5633.
[58] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[59] M. Althoff and G. Frehse, "Combining zonotopes and support functions for efficient reachability analysis of linear systems," in *Proc. IEEE Conf. Decis. Control*, 2016, pp. 7439–7446.
[60] F. Messerer and M. Diehl, "An efficient algorithm for tube-based robust nonlinear optimal control with optimal linear feedback," in *Proc. IEEE Conf. Decis. Control*, 2021, pp. 6714–6721.
[61] M. Wetzlinger, N. Kochdumper, S. Bak, and M. Althoff, "Fully-automated verification of linear systems using reachability analysis with support functions," in *Proc. Conf. Hybrid Syst.: Comput. Control*, 2023, pp. 1–12.
[62] V. Gaßmann and M. Althoff, "Scalable zonotope-ellipsoid conversions using the Euclidean zonotope norm," in *Proc. Amer. Control Conf.*, 2020, pp. 4715–4721.
[63] A. B. Kurzhanski and P. Varaiya, "On ellipsoidal techniques for reachability analysis. Part I: External approximations," *Optim. Methods Softw.*, vol. 17, no. 2, pp. 177–206, 2002.
[64] S. P. Han and O. L. Mangasarian, "Exact penalty functions in nonlinear programming," *Math. Program.*, vol. 17, no. 1, pp. 251–269, 1979.
[65] C. L. Guernic, "Reachability analysis of hybrid systems with linear continuous dynamics," Ph.D. dissertation, Université Joseph-Fourier - Grenoble I, Saint-Martin-d'Héres, France, 2009.
[66] H. Bell, "Gershgorins theorem and the zeros of polynomials," *Amer. Math. Monthly*, vol. 72, no. 3, pp. 292–295, 1965.
[67] N. Kochdumper, "Extensions of polynomial zonotopes and their application to verification of cyber-physical systems," Ph.D. dissertation, Technische Universität München, Munich, Germany, 2022.
[68] C. Cartis, N. I. M. Gould, and P. L. Toint, "On the evaluation complexity of constrained nonlinear least-squares and general constrained nonlinear optimization using second-order methods," *SIAM J. Numer. Anal.*, vol. 53, no. 2, pp. 836–851, 2015.
[69] Y. Li and J. Liu, "ROCS," in *Proc. Conf. Hybrid Syst.: Comput. Control*, 2018, pp. 130–135.
[70] Y. Li, Z. Sun, and J. Liu, "ROCS 2.0: An integrated temporal logic control synthesis tool for nonlinear dynamical systems," in *Proc. Conf. Anal. Des. Hybrid Syst.*, 2021, pp. 31–36.
[71] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," in *Proc. IEEE Conf. Decis. Control*, 2008, pp. 4042–4048.
[72] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2005.
[73] M. Althoff, "An introduction to CORA 2015," in *Proc. Workshop Appl. Verification Continuous Hybrid Syst.*, 2015, pp. 120–151.
[74] F. Immler, "Verified reachability analysis of continuous systems," in *Tools and Algorithms for the Construction and Analysis of Systems*. Berlin, Germany: Springer, 2015, pp. 37–51.
[75] N. Kochdumper, F. Gruber, B. Schürmann, V. Gaßmann, M. Klischat, and M. Althoff, "AROC: A toolbox for automated reachset optimal controller synthesis," in *Proc. Conf. Hybrid Syst.: Comput. Control*, 2021, pp. 1–6.
[76] L. Geretti et al., "ARCH-COMP22 category report: Continuous and hybrid systems with nonlinear dynamics," in *Proc. Workshop Appl. Verification Continuous Hybrid Syst.*, 2022, pp. 86–112.
[77] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus With Applications in Statistics and Econometrics*. Hoboken, NJ, USA: Wiley, 2019.
[78] A. J. Laub, *Matrix Analysis for Scientists and Engineers*, vol. 91. Philadelphia, PA, USA: SIAM, 2005.

VICTOR GAßMANN received the B.Sc. and M.Sc. degrees in electrical engineering and information technology from the Technical University of Munich, Munich, Germany, in 2016 and 2019, respectively. In 2019, he joined the Cyber-Physical Systems Group as a Ph.D. Student under the supervision of Prof. Dr.-Ing. Matthias Althof. His research interests include using optimization theory and reachability analysis for formally verified and robust controller synthesis of nonlinear systems.

MATTHIAS ALTHOFF is currently an Associate Professor of computer science with Technische Universität München, Munich, Germany. He received the Diploma in mechanical engineering and the Ph.D. degree in electrical engineering from the Technical University of Munich, Munich, Germany, in 2005 and 2010, respectively. From 2010 to 2012, he was a Postdoctoral Researcher with Carnegie Mellon University, Pittsburgh, PA, USA. From 2012 to 2013, he was an Assistant Professor with Technische Universität Ilmenau, Ilmenau, Germany. His research interests include formal verification of continuous and hybrid systems, reachability analysis, planning algorithms, nonlinear control, automated vehicles, robotics, and power systems.