Robust Optimal Control Using Set-based Reachability Analysis

Lukas Schäfer and Matthias Althoff

***Abstract*— Providing formal guarantees for robust constraint satisfaction is a crucial problem in safety-critical applications. We address this problem by combining robust optimal control with set-based reachability analysis of nonlinear systems. Our algorithm is built on a recently proposed scheme for optimizing affine feedback policies by iterating between solving a perturbed nominal optimal control problem and synthesizing linear feedback controllers – however, robust constraint satisfaction is not guaranteed. To overcome this limitation, we leverage concepts from robust controller synthesis and set-based reachability analysis. Our experiments show only a modest decrease in performance compared to an approximately robust approach, while we outperform state-of-the-art approaches that guarantee robust constraint satisfaction.**

I. Introduction

Robust nonlinear optimal control is a promising technique for planning and control in safety-critical applications: Since trajectories are generated by solving an optimization problem, constraints on the state and control input can be explicitly considered. Formal guarantees for robust constraint satisfaction can be provided despite disturbances, see e.g., [1], [2], but these guarantees usually come at the cost of compromised control performance.

Robust optimal control problems (OCPs) have mostly been studied in the context of robust model predictive control (MPC), where tube-based MPC has emerged as the most popular technique: Instead of a single trajectory, tube MPC approaches compute reachable sets, usually referred to as tube or funnel. The reachable sets are guaranteed to contain the true system state for every disturbance trajectory with values constrained within a compact set. To mitigate the growth of the reachable sets, tube MPC approaches include a feedback controller.

In rigid tube MPC, the feedback controller and a robust positively invariant set are computed offline using, e.g., incremental Lyapunov functions [1], [3] or control contraction metrics [4], [5]. Using such an invariant set as a rigid tube around a nominal trajectory yields a perturbed nominal OCP and, thus, the online computational complexity is roughly the same as for nominal MPC. However, designing the feedback controller and, hence, the tube offline can introduce conservatism.

To address this issue, the feedback controller can be chosen as an optimization variable in the robust OCP: Ellipsoidal tubes are directly predicted in [6], but this approach requires several auxiliary variables that are difficult to initialize. Polytopic tubes are usually less conservative than ellipsoidal tubes and enable a flexible interpolation-based control law [7] that results in a large number of optimization variables. For linear systems with additive disturbances, the robust OCP can be formulated as a single convex program by leveraging affine disturbance feedback [8] or system level synthesis [9], [10]. This approach has recently been extended to optimizing affine feedback policies of nonlinear systems [11]. However, the parameterization of the control law, again, leads to a large number of optimization variables.

Iterating between optimizing a nominal trajectory and synthesizing linear feedback controllers aims at reducing the computational burden of optimizing affine feedback policies online. However, balancing the minimization of the size of the funnel and the required control effort as in [12] is not necessarily optimal. In [13], the decoupling of the robust OCP is derived via its first-order necessary conditions and it is shown that the optimal linear feedback gains follow as the solution of a finite-horizon stochastic linear quadratic regulator (LQR) problem. However, the ellipsoidal disturbance sets are only propagated approximately for reachability analysis, in a way that is commonly used to formulate stochastic OCPs [14], [15]. Thus, robust constraint satisfaction cannot be guaranteed. Recently, this approach has been leveraged to reduce the computational burden in system level synthesis-based OCP formulations [16].

Contribution and Outline: In this paper, we propose a novel approach for solving robust nonlinear OCPs with affine feedback policies as optimization variables that iterates between solving perturbed nominal OCPs and optimizing the tube. In particular, we

- improve the performance compared to [12] by automatically tuning the cost function for the feedback controller synthesis based on the perturbed OCP solution [13].
- synthesize linear feedback controllers for the nonlinear error dynamics using concepts from linear matrix inequality (LMI)-based robust controller synthesis; and
- formally guarantee robust constraint satisfaction by leveraging set-based reachability analysis. In contrast, the linearization error is neglected and ellipsoidal disturbance sets are only propagated approximately in [13].

The remainder of this paper is structured as follows: In Sec. II, we introduce some preliminaries and provide our problem statement.. We present our approach for solving robust OCPs in Sec. III and evaluate it in Sec. IV.

Notation: The sets of natural numbers with and without zero are denoted by $\mathbb{N}_0$ and $\mathbb{N}$, respectively, and the set $\{r, r+1, \dots, q\} \subset \mathbb{N}_0$ is denoted by $\mathbb{N}_{[r:q]}$. The matrix full of ones and zeros of appropriate dimension are denoted by $\mathbf{1}$ and $\mathbf{0}$, respectively, and the identity matrix of dimension n is

The authors are with the TUM School of Computation, Information, and Technology, Technical University of Munich, Boltzmannstr. 3, 85748 Garching b. München, Germany. Email: {lukas.schaefer, althoff}@tum.de

denoted by I_n. The absolute value $|A|$ as well as equalities and inequalities between vectors and matrices are evaluated elementwise. For $a \in \mathbb{R}^n$, the operator $\operatorname{diag}(a)$ returns a diagonal matrix with the elements of a on the main diagonal. Similary, for a finite set of matrices $A_1, \ldots, A_m$, where $A_i \in \mathbb{R}^{n_i \times n_i}$, $\operatorname{diag}(A_1, \ldots, A_m)$ returns a block-diagonal matrix with the matrices $A_1, \ldots, A_m$ on the main diagonal. Given a vector-valued function $f(x)$, $\nabla f(x)$ denotes the transpose of the Jacobian of f. In the case of a vector-valued function $g(x, y)$ with multiple input arguments, we specify the arguments $\nabla_x g(x, y)$ if we do not take the Jacobian with respect to (w.r.t.) all arguments. The support function of a compact convex set $\mathcal{A} \subset \mathbb{R}^n$ in the direction of $l \in \mathbb{R}^n$ is denoted by $\rho_{\mathcal{A}}(l)$ [17, Sec. 8.1.3]. Given two sets $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^n$, $\mathcal{A} \times \mathcal{B}$ denotes their Cartesian product and $\mathcal{A} \oplus \mathcal{B}$ denotes their Minkowski sum. For matrix products of the form $A^T B A$, we sometimes use $[\bullet]^T B A$ for readability.

II. Preliminaries and Problem Statement

A. Preliminaries

We consider perturbed discrete-time nonlinear systems of the form

$$x_{k+1} = f(x_k, u_k, w_k), \tag{1}$$

where $x_k \in \mathbb{R}^{n_x}$ is the system state, $u_k \in \mathbb{R}^{n_u}$ is the control input, and $w_k \in \mathbb{R}^{n_w}$ is the unknown disturbance at time $k\Delta t$ for time step $k \in \mathbb{N}_{[0:N-1]}$ and sampling time period $\Delta t > 0$. The nonlinear function f is assumed to be three times continuously differentiable. Moreover, w_k is always confined to the disturbance set $\mathcal{W} \subset \mathbb{R}^{n_w}$, i.e., $\forall k\colon w_k \in \mathcal{W}$. We assume that $\mathcal{W} = \left\{ w\colon \|w\|_p \leq 1 \right\}$ for $p \in \{2, \infty\}$. Please note that this assumption entails disturbance sets of the form $\left\{ w\colon w = c + G\omega, \|\omega\|_p \leq 1 \right\}$ since the affine transformation can be considered as part of the dynamics function f in (1).

Given the initial state x_{init} and a nominal input trajectory $\bar{u}_{(\cdot)}$, we define a nominal state trajectory $\bar{x}_{(\cdot)}$ as $\forall k \in \mathbb{N}_{[0:N-1]}\colon \bar{x}_{k+1} = f(\bar{x}_k, \bar{u}_k, \mathbf{0}), \bar{x}_0 = x_{\text{init}}$. To compensate for disturbances, we not only plan open-loop input trajectories $\bar{u}_{(\cdot)}$ but also consider linear feedback

$$\pi_k(x) = \bar{u}_k + K_k(x - \bar{x}_k), \tag{2}$$

with time-varying feedback gains $K_k \in \mathbb{R}^{n_u \times n_x}$.

Next, we define the closed-loop reachable set of the system in (1) under the control policy in (2).

Definition 1 (Reachable Set): For the system in (1), a set of initial states $\mathcal{X}_0 \subset \mathbb{R}^{n_x}$, a control policy π_0 of the form in (2), and a set of disturbances $\mathcal{W}$, the reachable set $\mathcal{R}^+(\mathcal{X}_0, \pi_0)$ at the next time step is

$$\mathcal{R}^+(\mathcal{X}_0, \pi_0) = \{ f(x_0, \pi_0(x_0), w_0) : x_0 \in \mathcal{X}_0, w_0 \in \mathcal{W} \}. \tag{3}$$

By applying (3) recursively starting from $\mathcal{X}_0 = \{x_{\text{init}}\}$, we obtain the reachable set at time step k denoted by $\mathcal{R}\left(k, \pi_{(\cdot)}\right)$.

Since the exact computation of reachable sets of general nonlinear systems is impossible [18], we compute over-approximations, i.e., $\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right) \supseteq \mathcal{R}\left(k, \pi_{(\cdot)}\right)$ to ensure robust constraint satisfaction. We compute $\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right)$ using the discrete-time version of the reachability algorithm in [19]:

$$\begin{aligned} \widehat{\mathcal{R}}\left(k+1, \pi_{(\cdot)}\right) = \{\bar{x}_{k+1}\} \oplus (A_k + B_k K_k) \Delta\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right) \\ \oplus E_k \mathcal{W} \oplus \mathcal{L}(k), \\ \text{with } \widehat{\mathcal{R}}\left(0, \pi_{(\cdot)}\right) = \{x_{\text{init}}\}, \end{aligned} \tag{4}$$

where A_k, B_k, and E_k denote the first-order derivatives of f evaluated along the nominal trajectory and the Lagrange remainder $\mathcal{L}(k)$ encloses the set of linearization errors. We define the reachable set of the error dynamics as $\Delta\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right) = \widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right) \oplus \{-\bar{x}_k\}$. In the case of an ellipsoidal disturbance set $\mathcal{W}$, we compute a zonotopic enclosure for reachability analysis [20].

The state and the control input are constrained by

$$\forall k \in \mathbb{N}_{[0:N]}\colon x_k \in \mathcal{X}_k, \tag{5a}$$

$$\forall k \in \mathbb{N}_{[0:N-1]}\colon \pi_k(x_k) \in \mathcal{U}_k, \tag{5b}$$

where all $\mathcal{U}_k$ are assumed to be compact. Moreover, we assume that there exist $n_{h,k}$ twice continuously differentiable, scalar functions $h_{k,i}(x, u)$ with $\{(x, u)\colon h_{k,i}(x, u) \geq 0\}$ being convex so that the constraints in (5) can be equivalently expressed as

$$\forall k \in \mathbb{N}_{[0:N-1]}, \forall i \in \mathbb{N}_{[1:n_{h,k}]}\colon h_{k,i}(x_k, \pi_k(x)) \leq 0, \tag{6a}$$

$$\forall i \in \mathbb{N}_{[1:n_{h,N}]}\colon\ h_{N,i}(x_N) \leq 0. \tag{6b}$$

Figuratively speaking, this assumption implies that the sets of forbidden states and control inputs can be modeled as a finite union of convex keep-out zones. Thus, we cover a broad range of settings from the common polytopic admissible sets in MPC [2], [21], to collision avoidance constraints for autonomous systems in dynamic traffic scenarios [22], [23].

B. Problem statement

We consider robust OCPs of the form

$$\min_{\substack{\bar{x}_{(\cdot)}, \bar{u}_{(\cdot)}, \\ K_{(\cdot)}, \beta_{(\cdot)}}} \sum_{k=0}^{N-1} l(\bar{x}_k, \bar{u}_k) + V_T(\bar{x}_N) \tag{7a}$$

$$\text{s.t.}\quad \bar{x}_0 = x_{\text{init}}, \tag{7b}$$

$$\forall k \in \mathbb{N}_{[0:N-1]}\colon \bar{x}_{k+1} = f(\bar{x}_k, \bar{u}_k, 0), \tag{7c}$$

$$\forall k \in \mathbb{N}_{[0:N-1]}\colon h_k(\bar{x}_k, \bar{u}_k) + \sqrt{\beta_k + \epsilon} \leq 0, \tag{7d}$$

$$h_N(\bar{x}_N) + \sqrt{\beta_N + \epsilon} \leq 0, \tag{7e}$$

$$\begin{aligned} &\forall k \in \mathbb{N}_{[0:N-1]}, \forall i \in \mathbb{N}_{[1:n_{h,k}]}\colon \\ &\quad \beta_{k,i} = \rho_{\Delta\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right)}\left(\nabla h_{k,i}(\bar{x}_k, \bar{u}_k)\right)^2, \end{aligned} \tag{7f}$$

$$\begin{aligned} &\forall i \in \mathbb{N}_{[1:n_{h,N}]}\colon \\ &\quad \beta_{N,i} = \rho_{\Delta\widehat{\mathcal{R}}\left(N, \pi_{(\cdot)}\right)}\left(\nabla h_{N,i}(\bar{x}_N)\right)^2, \end{aligned} \tag{7g}$$

where the stage cost function $l(\cdot, \cdot)$ and terminal cost function $V_T(\cdot)$ are twice continuously differentiable, and $h_k(\bar{x}_k, \bar{u}_k)$, β_k, $h_N(\bar{x}_N)$, and β_N denote the vertical concatenations of all $h_{k,i}(\bar{x}_k, \bar{u}_k)$, $\beta_{k,i}$, $h_{N,i}(\bar{x}_N)$, and $\beta_{N,i}$, respectively. The constant $\epsilon \ll 1$ is added in (7d) and (7e) to ensure differentiability. We introduced the auxiliary variables β_k to leverage the approach from [13].

It remains to be shown that satisfaction of the constraints in (7d)-(7g) is sufficient for satisfaction of the constraints in (6). Inserting (7g) into (7e) yields

$$\begin{aligned} 0 &> h_{N,i}\left(\bar{x}_N\right) + \rho_{\Delta\widehat{\mathcal{R}}\left(N,\pi_{(\cdot)}\right)}\left(\nabla h_{N,i}\left(\bar{x}_N\right)\right) \\ &= h_{N,i}\left(\bar{x}_N\right) + \max_{\Delta x \in \Delta\widehat{\mathcal{R}}\left(N,\pi_{(\cdot)}\right)} \nabla h_{N,i}\left(\bar{x}_N\right)^T \Delta x \\ &\geq \max_{x \in \widehat{\mathcal{R}}\left(N,\pi_{(\cdot)}\right)} h_{N,i}\left(x\right), \end{aligned}$$

where the first step follows from the definition of the support function and the second step follows from the first-order condition for concavity of $h_{N,i}\left(x\right)$ [17, Sec. 3.1.3] (which follows from convexity of the zero-superlevel set). Hence, satisfaction of (7e) and (7g) is sufficient for satisfaction of (6b). Analogously, sufficiency of (7d) and (7f) for the satisfaction of (6a) can be shown.

III. Robust Optimal Control

We present our approach to address the robust OCP in (7), which adopts the core idea of the algorithm in [13]: the robust OCP is addressed by iterating between solving two subsets of its first-order necessary conditions – one subset corresponds to optimizing a nominal trajectory, the other to optimizing the feedback controllers. In contrast to [13], our approach ensures robust constraint satisfaction. Each iteration of our approach consists of the following steps (see Fig. 1):

1) **Solve perturbed OCP:** We formulate a perturbed nominal OCP by freezing the feedback gains $K_{(\cdot)}$ and, thus, the reachable sets $\Delta\widehat{\mathcal{R}}\left(\cdot,\pi_{(\cdot)}\right)$. We present the perturbed OCP and the computation of the perturbations in Sec. III-A.
2) **Synthesize feedback controllers:** We synthesize linear feedback controllers for the nonlinear error dynamics by solving a sequence of semi-definite programs, which we derive in Sec. III-B.
3) **Compute reachable sets:** Given the updated policies $\pi_{(\cdot)}^{(j+1)}$, we compute the reachable sets of the error dynamics $\Delta\widehat{\mathcal{R}}\left(\cdot,\pi_{(\cdot)}^{(j+1)}\right)$ using the procedure in (4) starting from the initial set $\Delta\widehat{\mathcal{R}}\left(0,\pi_{(\cdot)}^{(j+1)}\right) = \{\mathbf{0}\}$.
4) **Check constraints:** We check satisfaction of the constraints in (7d)-(7g) at the end of each iteration to ensure soundness.

We repeat these steps until the following simple convergence criterion is satisfied

$$\Delta_y = \left\| \begin{bmatrix} \bar{x}_{(\cdot)}^{(j+1)} \\ \bar{u}_{(\cdot)}^{(j+1)} \end{bmatrix} - \begin{bmatrix} \bar{x}_{(\cdot)}^{(j)} \\ \bar{u}_{(\cdot)}^{(j)} \end{bmatrix} \right\|_2 \leq \epsilon_y, \tag{8a}$$

$$\Delta_K = \left\| K_{(\cdot)}^{(j+1)} - K_{(\cdot)}^{(j)} \right\|_F \leq \epsilon_K, \tag{8b}$$

where $\epsilon_y \in \mathbb{R}_{>0}$, $\epsilon_K \in \mathbb{R}_{>0}$ are user-defined parameters. In the first iteration, we solve a nominal OCP, i.e., $\beta_{(\cdot)} = \mathbf{0}$ in (7), instead of the perturbed OCP. We discuss robust constraint satisfaction in Sec. III-C.

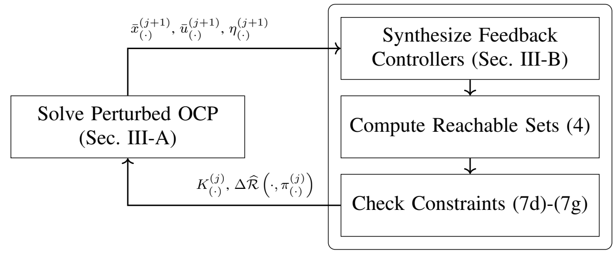

Fig. 1: Overview of our algorithm for solving the robust OCP in (7).

A. Perturbed Optimal Control Problem

We solve the following perturbed OCP to update the nominal state and input trajectory while freezing $K_{(\cdot)}$ and $\Delta\widehat{\mathcal{R}}\left(\cdot,\pi_{(\cdot)}\right)$:

$$\begin{aligned} &\left(\bar{x}_{(\cdot)}^{(j+1)}, \bar{u}_{(\cdot)}^{(j+1)}\right) = \\ &\quad \underset{\bar{x}_{(\cdot)},\bar{u}_{(\cdot)}}{\arg\min} \sum_{k=0}^{N-1} l\left(\bar{x}_k,\bar{u}_k\right) + \tilde{c}_k^T \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} + V_T\left(\bar{x}_N\right) + \tilde{c}_N^T \bar{x}_N \\ &\qquad + \kappa_y \left(\left\| \bar{x}_{(\cdot)} - \bar{x}_{(\cdot)}^{(j)} \right\|_2^2 + \left\| \bar{u}_{(\cdot)} - \bar{u}_{(\cdot)}^{(j)} \right\|_2^2 \right) \end{aligned} \tag{9a}$$

$$\text{s.t.} \quad \bar{x}_0 = x_{\text{init}}, \tag{9b}$$

$$\forall k \in \mathbb{N}_{[0:N-1]}: \bar{x}_{k+1} = f\left(\bar{x}_k,\bar{u}_k,0\right), \tag{9c}$$

$$\forall k \in \mathbb{N}_{[0:N-1]}: h_k\left(\bar{x}_k,\bar{u}_k\right) + b_k \leq 0, \tag{9d}$$

$$h_N\left(\bar{x}_N\right) + b_N \leq 0, \tag{9e}$$

where the constraint tightenings b_k in (9d) and b_N in (9e) are obtained by evaluating the constraints in (7f) and (7g), respectively, for the solution from the previous iteration.

The coefficients $\tilde{c}_k$ of the cost correction terms denote the gradients w.r.t. $\bar{x}_k$, $\bar{u}_k$ of (see [13, Sec. III-A])

$$\begin{aligned} \Phi = &\sum_{k=0}^{N-1} \sum_{i=1}^{n_{h,k}} \eta_{k,i} \rho_{\Delta\widehat{\mathcal{R}}\left(k,\pi_{(\cdot)}\right)} \left(\nabla h_{k,i}\left(\bar{x}_k,\bar{u}_k\right)\right)^2 \\ &+ \sum_{i=1}^{n_{h,N}} \eta_{N,i} \rho_{\Delta\widehat{\mathcal{R}}\left(N,\pi_{(\cdot)}\right)} \left(\nabla h_{N,i}\left(\bar{x}_N\right)\right)^2 \end{aligned} \tag{10}$$

evaluated at the solution from the previous (jth) iteration $\bar{x}_{(\cdot)}^{(j)}$, $\bar{u}_{(\cdot)}^{(j)}$, $K_{(\cdot)}^{(j)}$, and $\eta_{(\cdot)}^{(j)}$, where $\eta_{k,i}$ denote the Lagrange multipliers associated with the constraints in (7f), (7g). Thus, computing $\tilde{c}_k$ requires differentiating the algorithms for synthesizing the feedback controllers and computing the reachable sets w.r.t. to the input arguments $\bar{x}_{(\cdot)}$, $\bar{u}_{(\cdot)}$. Among others, this would require the computationally expensive evaluation of third-order tensors of the dynamics function f, see (4). For simplicity, we therefore propose to neglect the sensitivity of $K_{(\cdot)}$ w.r.t. $\bar{x}_{(\cdot)}$, $\bar{u}_{(\cdot)}$. To prevent the approximation error from growing too large, we add the penalty term to the cost function in (9a), which penalizes large deviations of $\bar{x}_{(\cdot)}, \bar{u}_{(\cdot)}$ from the solution of the previous iteration $\bar{x}_{(\cdot)}^{(j)}, \bar{u}_{(\cdot)}^{(j)}$. Since the proposed simplification corresponds to the case of fixed feedback gains (see also [24]), we introduce a similar penalty term when synthesizing $K_{(\cdot)}^{(j+1)}$ in Sec. III-B.2. Such penalty terms are regularly used in iterative algorithms for optimal control to facilitate convergence [12], [14].

To compute $\tilde{c}_k$, we use the differentiable approximation of the support function $\rho_{\Delta\widehat{\mathcal{R}}\left(k,\pi_{(\cdot)}\right)}(l)$ proposed in [25]: We compute the shape matrix Γ_k of an ellipsoidal approximation $\Delta\widetilde{\mathcal{R}}_{\mathcal{E}}\left(k,\pi_{(\cdot)}\right) \approx \Delta\widehat{\mathcal{R}}\left(k,\pi_{(\cdot)}\right)$ by propagating the ellipsoidal approximation $\{w : \|w\|_2 \leq 1\}$ of $\mathcal{W}$, i.e.,

$$\begin{aligned}\Gamma_{k+1} &= \left(A_k + B_k K_k\right)^T \Gamma_k \left(A_k + B_k K_k\right) + E_k^T E_k,\\ &\text{with } \Gamma_0 = \mathbf{0},\end{aligned}$$

where the Minkowski sum in (4) is approximated by adding the shape matrices. Using Γ_k, the approximation of the support function is defined as [25]

$$\begin{aligned}\tilde{\rho}_{\Delta\widehat{\mathcal{R}}\left(k,\bar{u}_{(\cdot)},K_{(\cdot)}^{(j)}\right)}(l) &= \frac{\rho_{\Delta\widehat{\mathcal{R}}\left(k,\pi_{(\cdot)}^{(j)}\right)}(l)}{\rho_{\Delta\widetilde{\mathcal{R}}_{\mathcal{E}}\left(k,\pi_{(\cdot)}^{(j)}\right)}(l)} \rho_{\Delta\widetilde{\mathcal{R}}_{\mathcal{E}}\left(k,\bar{u}_{(\cdot)},K_{(\cdot)}^{(j)}\right)}(l),\\ \text{with } \rho_{\Delta\widetilde{\mathcal{R}}_{\mathcal{E}}\left(k,\bar{u}_{(\cdot)},K_{(\cdot)}\right)}(l) &= \sqrt{l^T \Gamma_k l + \epsilon},\end{aligned}$$

where $\rho_{\Delta\widetilde{\mathcal{R}}_{\mathcal{E}}\left(k,\bar{u}_{(\cdot)},K_{(\cdot)}^{(j)}\right)}(l)$ is scaled to ensure that the approximation of the support function is exact for the current candidate policy $\pi_{(\cdot)}^{(j)}$, i.e.,

$$\tilde{\rho}_{\Delta\widehat{\mathcal{R}}\left(k,\bar{u}_{(\cdot)}^{(j)},K_{(\cdot)}^{(j)}\right)}(l) = \rho_{\Delta\widehat{\mathcal{R}}\left(k,\bar{u}_{(\cdot)}^{(j)},K_{(\cdot)}^{(j)}\right)}(l).$$

Note that we expanded the policy in the input arguments to indicate which parameters of the policy are (not) fixed. Inserting $\tilde{\rho}_{\Delta\widehat{\mathcal{R}}\left(k,\bar{u}_{(\cdot)},K_{(\cdot)}^{(j)}\right)}(l)$ into (10) enables the computation of $\tilde{c}_k \approx \nabla_{(x,u)}\Phi$.

By freezing the feedback gains $K_{(\cdot)}$ and the reachable sets $\Delta\widehat{\mathcal{R}}\left(\cdot,\pi_{(\cdot)}\right)$, the constraints in (7f), (7g) are eliminated from the perturbed OCP, compare (7) and (9). However, we require the Lagrange multipliers $\eta_{k,i}$ associated with these constraints in (10). Given the Lagrange multipliers $\lambda_{(\cdot)}^{(j+1)}$ associated with the tightened constraints in (9d), (9e), we obtain $\eta_{k,i}$ by evaluating (see [13, Sec. III-A])

$$\eta_{k,i}^{(j+1)} = \frac{1}{2b_{k,i}} \lambda_{k,i}^{(j+1)}. \tag{11}$$

B. Feedback Controller Synthesis

The crucial step for decomposing the robust OCP is interpreting the gradient of the Lagrangian w.r.t. $K_{(\cdot)}$ as the first-order necessary condition of (see [13, Sec. III-A])

$$K_{(\cdot)}^{(j+1)} = \underset{K_{(\cdot)}}{\arg\min}\ \Phi\left(\bar{x}_{(\cdot)}^{(j+1)}, \bar{u}_{(\cdot)}^{(j+1)}, K_{(\cdot)}, \eta_{(\cdot)}^{(j+1)}\right). \tag{12}$$

Figuratively speaking, this optimization problem returns a sequence of feedback gains $K_{(\cdot)}$ that minimizes the size of the reachable set $\Delta\widehat{\mathcal{R}}\left(\cdot,\pi_{(\cdot)}\right)$, measured via the support functions $\rho_{\Delta\widehat{\mathcal{R}}\left(k,\pi_{(\cdot)}\right)}(\cdot)$, in the direction of the active constraints (indicated by $\eta_{k,i} > 0$ assuming strict complementarity).

Directly solving the optimization problem in (12) requires optimization over $\Delta\widehat{\mathcal{R}}\left(k,\pi_{(\cdot)}\right)$, which is computationally expensive. In the approximately robust setting in [13], solving (12) is equivalent to solving a finite-horizon discrete-time stochastic LQR problem. To preserve the dynamic programming structure of the finite-horizon stochastic LQR problem and, thus, to reduce the computational burden, we rewrite the optimization problem in (12) as

$$\begin{aligned}\min_{K_{(\cdot)}} \max_{w_{(\cdot)}\in\mathcal{W}} \quad & \sum_{k=0}^{N-1} \Delta x_k^T \begin{bmatrix} I_{n_x} & K_k^T \end{bmatrix} \begin{bmatrix} Q_k & \mathbf{0} \\ \mathbf{0} & R_k \end{bmatrix} \begin{bmatrix} I_{n_x} \\ K_k \end{bmatrix} \Delta x_k \\ & + \Delta x_N^T Q_N \Delta x_N && \text{(13a)}\\ \text{s.t.} \quad & \Delta x_0 = 0, && \text{(13b)}\\ & \forall k \in \mathbb{N}_{[0:N-1]} : \Delta x_{k+1} = \left(A_k + B_k K_k\right)\Delta x_k \\ & \qquad + E_k w_k + p_k, && \text{(13c)}\\ & \qquad p_k = \phi\left(\Delta x_k, K_k \Delta x_k, w_k\right), && \text{(13d)}\end{aligned}$$

where the cost matrices are defined as

$$\begin{aligned}\begin{bmatrix} Q_k & \mathbf{0} \\ \mathbf{0} & R_k \end{bmatrix} &= \nabla h_k\left(\bar{x}_k, \bar{u}_k\right) \operatorname{diag}\left(\eta_k\right) \nabla h_k\left(\bar{x}_k, \bar{u}_k\right)^T,\\ Q_N &= \nabla h_N\left(\bar{x}_N\right) \operatorname{diag}\left(\eta_N\right) \nabla h_N\left(\bar{x}_N\right)^T,\end{aligned}$$

and the function ϕ captures the linearization error.

Instead of minimizing over a sequence of control inputs $u_{(\cdot)}$, a solution to problems of the form in (13) can be obtained by solving the Bellman equation for $k = N-1, \ldots, 0$:

$$\begin{aligned}V_k(x) &= \min_u \max_{w\in\mathcal{W}}\ l(x,u) + V_{k+1}\left(f(x,u,w)\right),\\ &\text{with } V_N(x) = V_T(x),\end{aligned} \tag{14}$$

where $V_k(x)$ denotes the value function. Differential dynamic programming [26] approximates this procedure by iterating between solving quadratic approximations of the Bellman equation (backward pass) and computing a new trajectory using the updated $u_{(\cdot)}$ (forward pass). Based on [27], [28], we combine differential dynamic programming with concepts from robust controller synthesis to handle the maximization over w and the linearization error in (13d).

To formulate the quadratic approximations of the Bellman equation, the control policy in differential dynamic programming is restricted to affine feedback policies [26], [27]. We only perform the backward pass once, since we compute $\bar{u}_{(\cdot)}$ and $\bar{x}_{(\cdot)}$ (result of the forward pass) by solving the perturbed OCP in (9) and consider the linearized dynamics around $\left(\bar{x}_{(\cdot)}, \bar{u}_{(\cdot)}\right)$ in (13).

Backward pass: For $k = N-1, \ldots, 0$, we obtain the feedback gain $K_k^{(j+1)}$ by solving the following relaxation of the Bellman equation

$$\begin{aligned}0 \geq \min_K \max_{w\in\mathcal{W}}\ & \Delta x\left(Q_k + K^T R_k K\right)\Delta x\\ & + \widetilde{V}_{k+1}\left(\Delta x^+\right) - \widetilde{V}_k\left(\Delta x\right),\end{aligned} \tag{15}$$

where $\Delta x^+ = \left(A_k + B_k K\right)\Delta x + E_k w + p$ and the relaxation is due to the quadratic approximation of the value function

$$\begin{aligned}\widetilde{V}_k\left(\Delta x\right) &= \begin{bmatrix} 1 \\ \Delta x \end{bmatrix}^T \underbrace{\begin{bmatrix} s_{k,11} & s_{k,12} \\ s_{k,21} & S_{k,22} \end{bmatrix}}_{=:S_k} \begin{bmatrix} 1 \\ \Delta x \end{bmatrix},\\ &\text{with} \quad S_N = \operatorname{diag}\left(0, Q_N\right),\end{aligned} \tag{16}$$

where $S_{k,22}$ is a symmetric positive-definite matrix. We show how to approximate (15) as the solution of an LMI in Sec. III-B.1. In Sec. III-B.2, we use this LMI to formulate a semi-definite program approximating (15).

1) Recasting the Bellman Equation as an LMI: To arrive at a semi-definite programming approximation of (15), we have to avoid the nonlinear equality constraint in (13d) and the non-convex maximization over w in (15). To this end, we relax the constraint in (13d) with the quadratic inequality

$$\begin{bmatrix} p \\ y_p \end{bmatrix}^T \underbrace{\begin{bmatrix} M_{k,11} & \mathbf{0} \\ \mathbf{0} & M_{k,22} \end{bmatrix}}_{=:M_k} \begin{bmatrix} p \\ y_p \end{bmatrix} \geq 0, \quad \text{with} \quad y_p = (C + DK)\,\Delta x + Gw, \tag{17}$$

for a negative definite matrix $M_{k,11}$ and a positive definite matrix $M_{k,22} \in \mathbb{R}^{n_q \times n_q}$. The matrices C, D, and G are matrices full of ones and zeros that extract the entries of x, u, and w, which enter any nonlinear term in f. In Sec. III-B.3, we derive a suitable matrix M_k via the reachability algorithm from [19].

While this relaxation inevitably introduces conservatism, it enables us to leverage the S-procedure to formulate an LMI that is sufficient for satisfaction of the inequality in (15): Combining (17) and (15) entails that a feedback gain K is a feasible solution of the problem in (15) if it satisfies

$$0 \geq \Delta x \left(Q_k + K^T R_k K\right) \Delta x + \widetilde{V}_{k+1}\left(\Delta x^+\right) - \widetilde{V}_k\left(\Delta x\right), \tag{18}$$

robustly, i.e., for all $w \in \mathcal{W}$ and $\Delta x, p$ that satisfy (17). From the S-procedure [29, Sec. 2.6.3], it follows that (18) holds if there exist $\lambda_w > 0$ and $\lambda_p > 0$ so that

$$\begin{aligned} 0 \geq\; & \widetilde{V}_{k+1}\left(\Delta x^+\right) + \Delta x^T \left(Q_k + K^T R_k K\right) \Delta x - \widetilde{V}_k\left(\Delta x\right) \\ & + \lambda_p \begin{bmatrix} p \\ y_p \end{bmatrix}^T M_k \begin{bmatrix} p \\ y_p \end{bmatrix} + \lambda_w \Big(1 - \|w\|_2^2\Big), \end{aligned} \tag{19}$$

for all Δx, w, p. Thus, every K solving (19) is also a feasible solution of (15). Please note that this step requires approximating $\mathcal{W}$ by the ellipsoid $\{w : \|w\|_2 \leq 1\}$.

A matrix inequality that is sufficient for the condition in (19) and, thus, (18), is given in (20), which can be easily verified by multiplying $\begin{bmatrix} 1 & \Delta x^T & w^T & p^T \end{bmatrix}$ to both sides. From positive semi-definiteness of the Schur complement [17, Sec. A.5.5], we obtain that (20) holds if and only if the matrix P_k defined in (21) is positive semi-definite. Since Q_k and R_k might only be positive semi-definite, we add ϵI_{n_x} and ϵI_{n_u}, $\epsilon \ll 1$, to Q_k and R_k, respectively, before computing the respective square roots. We multiply both sides of P_k with $T = \operatorname{diag}\left(I_{n_x+1}, I_{n_w}, \lambda_p^{-1} I_{n_x}, I_{n_x+1}, I_{n_x}, I_{n_u}, \lambda_w, I_{n_q}\right)$, which yields the LMI $T^T P_k T \succeq 0$ that implies (18).

2) Computing the Feedback Gain and the Value Function Approximation: As in [27], our value function approximation, which we obtain by solving $T^T P_k T \succeq 0$, is an upper bound on the optimal value function. Therefore, we adopt the cost function from [27, Eq. (14)] in the following semi-definite program to obtain a possibly tight upper bound on the value function:

$$\min_{K, S_k, \lambda_w, \lambda_p^{-1}} \quad \operatorname{trace}\left(S_k\right) + \kappa_K \left\|K - K_k^{(j)}\right\|_F \tag{22a}$$

$$\text{s.t.} \quad T^T P_k T \succeq 0, \tag{22b}$$

$$\lambda_w > 0,\ \lambda_p^{-1} > 0, \tag{22c}$$

where $\kappa_K \in \mathbb{R}_{>0}$ is a user-defined weight. We add a soft penalty term to the cost function due to our simplified cost correction terms in (9): To support the assumption that the sensitivity of $\Phi\left(\bar{x}_{(\cdot)}^{(j)}, \bar{u}_{(\cdot)}^{(j)}, K_{(\cdot)}, \eta_{(\cdot)}^{(j)}\right)$ w.r.t. $K_{(\cdot)}$ is negligible (see Sec. III-A), we aim to compute feedback gains that are close to our current candidate solution $K_j^{(k)}$. Moreover, we have found empirically that the penalty term in (22a) helps to stabilize the active set, which is crucial for the algorithm to converge.

3) (Approximate) Quadratic Bound on the Linearization Error: We show how to approximate the matrix M_k in (17) using a symmetric Lagrange remainder $\mathcal{L}(k)$ (see (4))

$$\mathcal{L}(k) = \left\{x\colon\ |x| \leq \ell\left(\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right)\right)\right\}$$

where $\ell\left(\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right)\right) \in \mathbb{R}_{\geq 0}^{n_x}$ bounds the Lagrange remainder, see e.g. [19, Proposition 1]. By combining (4) with (13c) and (13d), it follows that

$$|p_k| \leq \left|\ell\left(\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right)\right)\right| \tag{23}$$

holds. We use the approach from [19, Proposition 1] for computing $\ell\left(\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right)\right)$, i.e.,

$$\ell_i\left(\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right)\right) = \tfrac{1}{2} \Delta z^T H_i \Delta z,$$

where Δz collects the edge lengths of $\mathcal{Z}$, which denotes the smallest box enclosure of $\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right) \times K\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right) \times \mathcal{W}$, and $H_i = \max_{z \in \mathcal{Z}} \left|\nabla^2 f_i(z)\right|$. To obtain a quadratic bound on p as in (17), we use the following approximation of (23):

$$\begin{aligned} p^T p \leq & \left\| \underbrace{\begin{bmatrix} 1/2 \Delta z^T H_1 \\ \vdots \\ 1/2 \Delta z^T H_{n_x} \end{bmatrix}}_{=:\hat{H}} \begin{bmatrix} \Delta x \\ K\Delta x \\ w \end{bmatrix} \right\|_2^2 \\ = & \left\| \hat{H}_x \Delta x + \hat{H}_u K \Delta x + \hat{H}_w w \right\|_2^2 \\ \leq & \; 3 \left\|\hat{H}_x\right\|_F^2 \|\Delta x\|_2^2 + 3 \left\|\hat{H}_u\right\|_F^2 \|K\Delta x\|_2^2 \\ & + 3 \left\|\hat{H}_w\right\|_F^2 \|w\|_2^2, \end{aligned}$$

where the last step follows from Jensen's inequality [17, Sec. 3.1.8] and $\hat{H}_x$, $\hat{H}_u$, and $\hat{H}_w$ denote the respective columns of $\hat{H}$. Hence, we obtain $M_{k,11} = -I_{n_x}$ and

$$M_{k,22} = \operatorname{diag}\left(3\left\|\hat{H}_x\right\|_F^2 I_{n_x}, 3\left\|\hat{H}_u\right\|_F^2 I_{n_u}, 3\left\|\hat{H}_w\right\|_F^2 I_{n_w}\right).$$

$$0 \succeq [\bullet]^T \begin{bmatrix} -s_{k,11} & -s_{k,12} & & & & & & & & \\ -s_{k,21} & -S_{k,22} & & & & & & & & \\ & & -\lambda_w I_{n_w} & & & & & & & \\ & & & \lambda_p M_{k,11} & & & & & & \\ & & & & s_{k+1,11} & s_{k+1,12} & & & & \\ & & & & s_{k+1,21} & S_{k+1,22} & & & & \\ & & & & & & Q_k & & & \\ & & & & & & & R_k & & \\ & & & & & & & & \lambda_w & \\ & & & & & & & & & \lambda_p M_{k,22} \end{bmatrix} \begin{bmatrix} 1 & & & \\ & I_{n_x} & & \\ & & I_{n_w} & \\ & & & I_{n_x} \\ 1 & & & \\ & A_k + B_k K & E_k & I_{n_x} \\ & I_{n_x} & & \\ & K & & \\ 1 & & & \\ & C + DK & G & \end{bmatrix} \tag{20}$$

$$P_k = \begin{bmatrix} s_{k,11} & s_{k,12} & & & 1 & & & & 1 & \\ s_{k,12}^T & S_{k,22} & & & & (A_k + B_k K)^T & \left(Q_k^{1/2}\right)^T & \left(R_k^{1/2} K\right)^T & & (C + DK)^T \\ & & \lambda_w I_{n_w} & & & E_k^T & & & & G^T \\ & & & -\lambda_p M_{k,11} & & I_{n_x}^T & & & & \\ 1 & & & & \tilde{s}_{k+1,11} & \tilde{s}_{k+1,12} & & & & \\ & (A_k + B_k K) & E_k & I_{n_x} & \tilde{s}_{k+1,21}^T & \tilde{S}_{k+1,22} & & & & \\ & Q_k^{1/2} & & & & & I_{n_x} & & & \\ & R_k^{1/2} K & & & & & & I_{n_u} & & \\ 1 & & & & & & & & \lambda_w^{-1} & \\ & C + DK & G & & & & & & & \lambda_p^{-1} M_{k,22}^{-1} \end{bmatrix} \tag{21}$$

with $\tilde{S}_k = S_k^{-1}$.

Computing the entries in $M_{k,22}$ requires the reachable set $\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}^{(j+1)}\right)$, which, in turn, requires the feedback gain $K_{(\cdot)}^{(j+1)}$, which requires $M_{k,22}$. To resolve this mutual dependency, we use $K_{(\cdot)}^{(j)}$ and $\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}^{(j)}\right)$, i.e., the solution from the previous iteration, for computing $M_{k,22}$. Thus, we recover the correct matrix $M_{k,22}$ at a stationary point of our algorithm. A similar idea has been used in [12] to estimate the Lipschitz constant of a nonlinear system. We use this approximation of the quadratic constraint in (17) for the numerical experiments in Sec. IV.

C. Robust Constraint Satisfaction

To formulate the perturbed OCP and synthesize the feedback controllers, we employ several approximations of reachable sets. Nevertheless, any solution that passes the constraint check at the end of each iteration ensures robust constraint satisfaction since we compute over-approximations of the reachable sets $\Delta\widehat{\mathcal{R}}\left(\cdot, \pi_{(\cdot)}^{(j)}\right)$ under the updated policy $\pi_{(\cdot)}^{(j)}$. Please note that the computational overhead is negligible since we can re-use the computed support functions $\rho_{\Delta\widehat{\mathcal{R}}\left(k, \pi_{(\cdot)}\right)}\left(\nabla h_{k,i}\left(\bar{x}_k, \bar{u}_k\right)\right)$ for the formulation of the perturbed OCP in the next iteration.

If we embed the proposed robust optimal control algorithm in an MPC scheme, we can ensure recursive feasibility by employing a polytopic robust control invariant set Ω with an invariance-preserving affine policy π_Ω as a terminal constraint set: Assuming a feasible solution at the initial time step k_0 - as is standard in MPC - we can always obtain a safe policy by appending π_Ω to the safe policy from the previous time step. Note that we require a polytopic set Ω to formulate the constraint $\widehat{\mathcal{R}}\left(N, \pi_{(\cdot)}\right) \subseteq \Omega$ in the form of (7e),(7g).

IV. Numerical Experiments

In this section, we apply our approach to a mass-spring damper system, a unicycle, and a planar quadrotor collision avoidance task. We use CORA [30] for reachability analysis, IPOPT [31] via the MATLAB interface from the OPTI toolbox[1] for solving (9), CasADi [32] for computing derivatives, YALMIP [33] for modeling and Mosek [34] for solving (22).

We compare our approach with

1) the approximately robust approach proposed in [13], which optimizes over affine feedback policies;
2) designing the matrices Q_k, R_k in the cost function in (13a) for synthesizing the feedback controllers offline, which corresponds to the approach in [12]; and
3) open-loop robust optimal control, i.e., $K_{(\cdot)} = \mathbf{0}$, see e.g., [24].

In the second case, we chose the cost matrices in (13a) as $Q_k = I_{n_x}$, $R_k = 0.1 I_{n_u}$, and $Q_N = 10 I_{n_x}$. Table I provides the relative nominal costs, i.e., we excluded possible cost correction terms for comparison. For the same reason, we normalized the costs w.r.t. the cost achieved using the approach in [13]. To create a more challenging control problem, we not only considered the ellipsoidal disturbance sets from [6] and [12], respectively, (sixth column in Table I) but also the respective interval enclosures as the disturbance set $\mathcal{W}$ (seventh column in Table I). In the mass-spring-damper system experiment, the approach from [13] oscillated between a feasible and an infeasible solution due to an unstable active set. For comparison, we picked the cost for the feasible solution. In the open-loop case, no feasible solution was found, as the reachable set grew too large.

Comparing the performance of our approach and [13] to the case of fixed cost matrices Q_k, R_k in (13a) as well

[1] `https://github.com/jonathancurrie/OPTI`

TABLE I: Comparison of costs for three benchmarks, normalized w.r.t the approach in [13].

Experiment	n_x	n_u	n_w	[13]	Our approach $\|w\|_2 \leq 1$	Our approach $\|w\|_\infty \leq 1$	Fixed Q_k, R_k $\|w\|_2 \leq 1$	Open-loop $\|w\|_2 \leq 0.5$
Mass-spring-damper system (based on [6])	2	1	2	1	1.008	1.019	1.133	-
Unicycle (based on [12])	3	2	2	1	1.009	1.011	1.042	1.104
Planar quadrotor (based on [5])	6	2	1	1	1.008	1.008	1.051	223

as the open-loop robust case demonstrates the efficacy of (automatically) tuning the cost function for the feedback controller synthesis: by minimizing the size of the reachable set in the direction of the active constraints, lower costs can be achieved. As expected, propagating the disturbances approximately and neglecting the linearization error as in [13] enables a more aggressive controller, which results in the lowest cost throughout all three examples. However, our approach only results in a modest increase of the cost compared to [13], while guaranteeing safety : To demonstrate the robust constraint satisfaction properties, we uniformly sampled 1000 disturbance trajectories from $\mathcal{W}$ and simulated the closed-loop system under the optimal policies. Our approach always satisfied the constraints, whereas the approach from [13] resulted in a constraint violation in 20.5% and 6.7% of the runs for the mass-spring-damper system and the unicycle, respectively.

We now take a closer look at the unicycle benchmark. The dynamics of the unicycle are governed by

$$\begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} u_1 \cos(\theta) + 0.1 w_1 \\ u_1 \sin(\theta) + 0.1 w_2 \\ u_2 \end{bmatrix}, \tag{24}$$

where r_x, r_y, and θ, denote the position and orientation, respectively, u_1 is the velocity, and u_2 is the angular velocity. The inputs are bounded by $|u_1| \leq 4\,\mathrm{m/s}$ and $|u_2| \leq 2.5\,\mathrm{rad/s}$. The prediction horizon is $N = 30$ with the sampling time $\Delta t = 0.1\,\mathrm{s}$. To obtain the discrete-time dynamics, we use one step of a fourth-order Runge-Kutta integrator with step size Δt, while keeping the control input u and the disturbance w constant. The control goal is to minimize the control effort, i.e., $l\left(\bar{x}_k, \bar{u}_k\right) = \|\bar{u}_k\|_2^2$, while avoiding the ellipsoidal obstacles, see Fig. 2. The initial state is $x_{\text{init}} = \mathbf{0}$ and the system must reach the target region $4.5\,\mathrm{m} \leq r_x, r_y \leq 5.5\,\mathrm{m}$, $|\theta| \leq 20\pi/180$ at $k = N$.

The resulting tube and the nominal trajectory for $\|w\|_\infty \leq 1$ are shown in Fig. 2. The unicycle can navigate close to the obstacles despite the uncertainties. At the final time step, the size of $\widehat{\mathcal{R}}\left(N, \pi_{(\cdot)}\right)$ in r_x-direction is minimized to reach the target region with minimum control effort.

TABLE II: Average computation times per iteration for the unicycle benchmark.

	Perturbed OCP (9)	Feedback Synthesis	Reachability Analysis
Our Approach	0.06 s	0.29 s	0.19 s
[13]	0.06 s	0.03 s	0.01 s

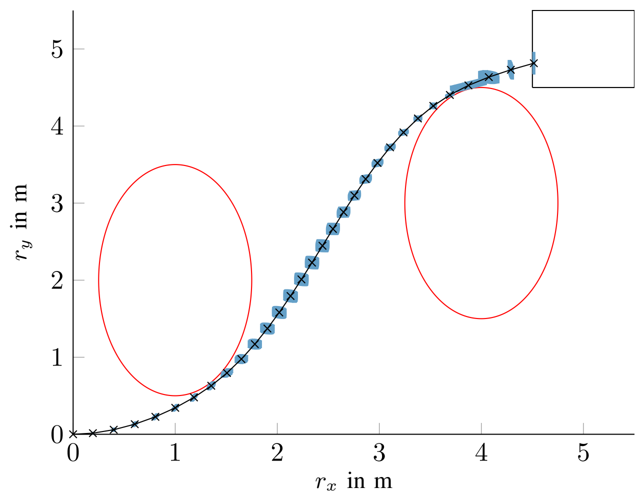

Fig. 2: Results for the unicycle benchmark with $\|w\|_\infty \leq 1$. Shown are: ellipsoidal obstacles (red), target region (black box), reachable sets (light blue), and nominal trajectory (black with x's).

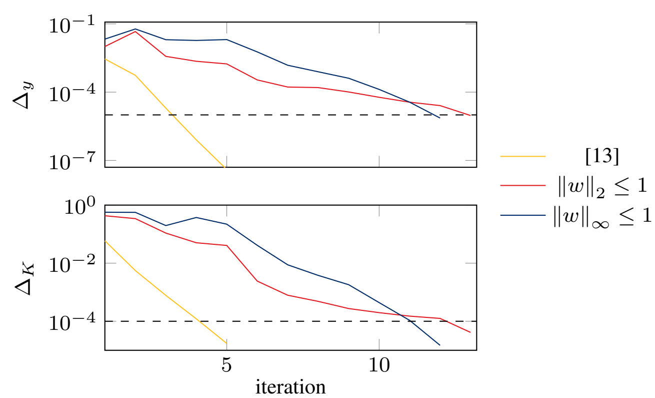

Fig. 3: Convergence plots for the unicycle benchmark. The dashed lines denote the thresholds $\epsilon_y = 10^{-5}$ and $\epsilon_K = 10^{-4}$ for the convergence criteria in (8a) and (8b), respectively.

The computation times and convergence plots for the approach from [13] and our approach are shown in Table II and Fig. 3, respectively. Solving the semi-definite program in (22), which is more expensive than the stochastic LQR in [13], (feedback synthesis) and the additional computation of the Lagrange remainder in (4) (reachability analysis) increase the computational effort compared to [13]. Promising directions to reduce the computational burden of our approach include developing structure-exploiting semi-definite programming solvers as demonstrated in [35]. For a better comparison of the convergence behavior, we consider the algorithm from [13] to have converged if the criteria in (8) are satisfied. The approach in [13] converges within a smaller number of iterations, see Fig. 3. Compared to [13], treating the disturbances robustly and accounting for the linearization error increases the level of uncertainty. As observed in [13], increasing the level of uncertainty slows down convergence.

V. Conclusion

We presented an algorithm that solves robust OCPs with affine feedback policies as optimization variables by iterating between optimizing the nominal trajectory and linear feedback controllers. To ensure robust constraint satisfaction, we leverage concepts from robust controller synthesis and set-based reachability analysis. The cost function for synthesizing the feedback controllers is tuned automatically based on the solution of the perturbed OCP to improve the performance. Our results demonstrate the improved performance compared to open-loop formulations and tuning the cost function for synthesizing the feedback controllers offline. We observe only a modest decrease in performance compared to an approximately robust approach, while guaranteeing robust constraint satisfaction.

ACKNOWLEDGMENTS

We gratefully acknowledge financial support by the project justITSELF funded by the European Research Council under grant agreement No 817629 and the German Federal Ministry for Economics Affairs and Climate Action project VaF under grant KK5135901KG0. Furthermore, we would like to thank Victor Gaßmann, Hanna Krasowski, Adrian Kulmburg, Mark Wetzlinger, and Alexander Wischnewski for fruitful discussions on the topic and on the manuscript.

References

[1] J. Köhler, R. Soloperto, M. A. Müller, *et al.*, "A Computationally Efficient Robust Model Predictive Control Framework for Uncertain Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 794–801, 2021.

[2] B. Schürmann, N. Kochdumper, and M. Althoff, "Reachset Model Predictive Control for Disturbed Nonlinear Systems," in *IEEE Conference on Decision and Control*, 2018, pp. 3463–3470.

[3] F. Bayer, M. Bürger, and F. Allgöwer, "Discrete-time Incremental ISS: A framework for Robust NMPC," in *European Control Conference*, 2013, pp. 2068–2073.

[4] S. Singh, A. Majumdar, J.-J. Slotine, *et al.*, "Robust online motion planning via contraction theory and convex optimization," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 5883–5890.

[5] P. Zhao, A. Lakshmanan, K. Ackerman, *et al.*, "Tube-Certified Trajectory Tracking for Nonlinear Systems With Robust Control Contraction Metrics," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5528–5535, 2022.

[6] M. E. Villanueva, R. Quirynen, M. Diehl, *et al.*, "Robust MPC via min–max differential inequalities," *Automatica*, vol. 77, pp. 311–321, 2017.

[7] M. E. Villanueva, M. A. Müller, and B. Houska, "Configuration-Constrained Tube MPC," *Automatica*, vol. 163, 111543, 2024.

[8] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski, "Optimization over state feedback policies for robust control with constraints," *Automatica*, vol. 42, no. 4, pp. 523–533, 2006.

[9] J. Anderson, J. C. Doyle, S. H. Low, *et al.*, "System level synthesis," *Annual Reviews in Control*, vol. 47, pp. 364–393, 2019.

[10] J. Sieber, S. Bennani, and M. N. Zeilinger, "A System Level Approach to Tube-Based Model Predictive Control," *IEEE Control Systems Letters*, vol. 6, pp. 776–781, 2022.

[11] A. P. Leeman, J. Köhler, A. Zanelli, *et al.*, *Robust nonlinear optimal control via system level synthesis*, 2024. arXiv: `2301.04943 [math.OC]`.

[12] T. Kim, P. Elango, and B. Açıkmeşe, "Joint synthesis of trajectory and controlled invariant funnel for discrete-time systems with locally lipschitz nonlinearities," *International Journal of Robust and Nonlinear Control*, vol. 34, no. 6, pp. 4157–4176, 2024.

[13] F. Messerer and M. Diehl, "An Efficient Algorithm for Tube-based Robust Nonlinear Optimal Control with Optimal Linear Feedback," in *IEEE Conference on Decision and Control*, 2021, pp. 6714–6721.

[14] T. Lew, R. Bonalli, and M. Pavone, "Chance-Constrained Sequential Convex Programming for Robust Trajectory Optimization," in *European Control Conference*, 2020, pp. 1871–1878.

[15] X. Feng, S. D. Cairano, and R. Quirynen, "Inexact Adjoint-based SQP Algorithm for Real-Time Stochastic Nonlinear MPC," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6529–6535, 2020.

[16] A. P. Leeman, J. Köhler, F. Messerer, *et al.*, "Fast System Level Synthesis: Robust Model Predictive Control using Riccati Recursions," in *IFAC Conference on Nonlinear Model Predictive Control*, 2024.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[18] A. Platzer and E. M. Clarke, "The Image Computation Problem in Hybrid Systems Model Checking," in *Hybrid Systems: Computation and Control*, Berlin, Heidelberg: Springer, 2007, pp. 473–486.

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

[20] V. Gaßmann and M. Althoff, "Scalable Zonotope-Ellipsoid Conversions using the Euclidean Zonotope Norm," in *American Control Conference*, 2020, pp. 4715–4721.

[21] J. Köhler, M. A. Müller, and F. Allgöwer, "A novel constraint tightening approach for nonlinear robust model predictive control," in *American Control Conference*, 2018, pp. 728–734.

[22] Y. Mao, D. Dueri, M. Szmuk, *et al.*, "Successive Convexification of Non-Convex Optimal Control Problems with State Constraints," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4063–4069, 2017.

[23] L. Schäfer, S. Manzinger, and M. Althoff, "Computation of Solution Spaces for Optimization-based Trajectory Planning," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 216–231, 2023.

[24] A. Zanelli, J. Frey, F. Messerer, *et al.*, "Zero-Order Robust Nonlinear Model Predictive Control with Ellipsoidal Uncertainty Sets," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 50–57, 2021.

[25] V. Gaßmann and M. Althoff, "Polynomial Controller Synthesis of Nonlinear Systems With Continuous State Feedback Using Trust Regions," *IEEE Open Journal of Control Systems*, vol. 2, pp. 310–324, 2023.

[26] D. Q. Mayne, "A Second-order Gradient Method for Determining Optimal Trajectories of Non-linear Discrete-time Systems," *International Journal of Control*, vol. 3, no. 1, pp. 85–95, 1966.

[27] D. Gramlich, C. W. Scherer, and C. Ebenbauer, "Robust Differential Dynamic Programming," in *IEEE Conference on Decision and Control*, 2022, pp. 1714–1721.

[28] D. Gramlich, C. W. Scherer, H. Häring, *et al.*, "Synthesis of constrained robust feedback policies and model predictive control," in *European Control Conference*, 2024, pp. 3476–3483.

[29] S. Boyd, L. El Ghaoui, E. Feron, *et al.*, *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.

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

[31] A. Wächter and L. T. Biegler, "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[32] J. A. E. Andersson, J. Gillis, G. Horn, *et al.*, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[33] J. Löfberg, "YALMIP : A Toolbox for Modeling and Optimization in MATLAB," in *In Proceedings of the CACSD Conference*, 2004.

[34] Mosek ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 10.0.25*, 2022.

[35] D. Gramlich, T. Holicki, C. W. Scherer, *et al.*, "A structure exploiting SDP solver for robust controller synthesis," *IEEE Control Systems Letters*, vol. 7, pp. 1831–1836, 2023.