Reachset Model Predictive Control for Disturbed Nonlinear Systems

Bastian Schürmann, Niklas Kochdumper, and Matthias Althoff

Abstract— **The popularity of model predictive control (MPC) is mainly founded on its easy implementation and its ability to consider state and input constraints. For future applications in safety-critical systems, however, it is necessary to provide formal guarantees of safety despite disturbances and measurement noise. In this paper, we include reachability analysis in an MPC approach to obtain provably safe controllers which are easy to implement. We consider continuous-time, nonlinear systems affected by disturbances and measurement noise. In contrast to most existing techniques, we explicitly consider the computation time and guarantee the satisfaction of state and input constraints despite the previously-mentioned disturbances. We use a novel type of dual mode MPC, which does not require the computation of Lyapunov functions. We demonstrate the applicability of our approach with a numerical example of a chemical reactor, where we show the advantages of our approach compared to existing MPC.**

I. Introduction

In the last decades, model predictive control has gained a lot of interest, both in academia as well as in industry. Advantages of this control method include its ability to provide optimized control trajectories and to handle constraints for both states and inputs. This, and its rather easy implementation, are the main reasons for its popularity in industry. With growing interest in safety-critical applications, such as autonomous driving or robots working together with humans, current techniques for model predictive control have to be enhanced to provide formal guarantees of correct behavior despite complex dynamics, limited actuation capabilities, external disturbances, and sensor noise.

Different techniques are proposed to achieve the above-mentioned goals, both for linear [5] and nonlinear systems [7], [21]: Many approaches optimize the center trajectory and use a second auxiliary controller to track it, thereby keeping the system in a "tube" around this trajectory. For linear systems, this works well due to the superposition principle [14], [19]. For nonlinear systems, this is harder, and different auxiliary controllers have been proposed, e.g., sliding mode controllers [27], dynamic games [17], or even a second MPC controller [18]. Recent approaches suggest the application of contraction theory to obtain a controller [29]. Other developments are parametrized [25] and homothetic MPC [26], which can adapt the tube size. Some approaches use interval arithmetic [16] and reachable sets based on zonotopes [6] to compute the reachable sets online.

This work is partially supported by the European Commission under the project UnCoVerCPS (grant number 643921) and by the German Research Foundation (DFG) project faveAC (grant number AL 1185/5-1).

The authors are with the Department of Informatics, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany (e-mail: {bastian.schuermann, niklas.kochdumper, althoff}@tum.de)

In contrast to regular MPC, which requires most computations and optimizations during runtime, the majority of these are performed offline in explicit MPC. This works for linear [1], [10] and for nonlinear systems [12], [22], even with disturbances [24], by dividing the state space into different partitions and computing a (sub-)optimal controller which satisfies the constraints offline in advance.

While a number of different approaches already exist, the question of how to obtain efficient and provably-safe MPC for constrained and disturbed nonlinear systems remains open. Due to the implicit nature of regular MPC, it is often not possible to apply formal verification tools like reachability analysis [2], [4], [8] in order to prove safety. While some approaches compute reachable sets online, they face the problem that their reachability analysis methods produce large over-approximations. That is also the case for other techniques with implicit safety guarantees, e.g., the contraction-based approach [29], which easily becomes rather conservative, since the contraction set has to hold everywhere in the considered state space. Explicit MPC, on the other hand, does not have these problems. However, due to the division of the state space, the computation scales exponentially with the number of dimensions and constraints, so that only small-dimensional systems can be considered. Another problem which most MPC approaches face is that most of them (with the exception of, e.g., [30]) do not consider computation time, therefore neglecting the fact that the time which is needed to perform the optimizations after a new measurement leads to delays and therefore possibly unsafe and unstable behavior. From a practical point of view, it is also often a problem that most techniques require a Lyapunov function in order to prove stability or to compute invariant sets. For real world, complex systems, Lyapunov functions are often hard to obtain.

The goal of this work is to combine reachability analysis with regular MPC in order to obtain provably safe controllers for disturbed nonlinear systems with constraints on states and inputs. Our new approach aims at transferring safety guarantees from reachability analysis to model predictive control. As a consequence, operators are not required to know any Lyapunov functions, nor have any other deep knowledge of control theory. This makes our approach particularly appealing for problems in practice. We are even able to consider the delay caused by the computation time of our approach. We also do not need to pre-compute a fixed tube size as required in many previously-mentioned approaches, which reduces conservatism. In addition, we are able to take continuous time dynamics and measurement noise, which are neglected in many existing approaches but which are critical

to provide safety guarantees, into account.

The paper is organized as follows. After a formal problem formulation in Sec. II, we present the main algorithm in Sec. III. The applicability of the algorithm is demonstrated in a numerical example in Sec. IV. This is followed by a discussion of the algorithm in Sec. V and a conclusion in Sec. VI.

II. Problem Formulation

We consider a continuous-time system with disturbed, nonlinear dynamics of the form

$$\dot{x}(t) = f\big(x(t), u(t), w(t)\big), \tag{1}$$

with states $x(t) \in \mathbb{R}^n$, inputs $u(t) \in \mathbb{R}^m$, and disturbances $w(t) \in \mathcal{W} \subset \mathbb{R}^d$ ($\mathcal{W}$ is compact, i.e., closed and bounded). We do not require any stochastic properties for $w(\cdot)$; we only assume that any possible disturbance trajectory is bounded at any point in time in the compact set $\mathcal{W}$. We denote this by $w(\cdot) \in \mathcal{W}$, which is shorthand for $w(t) \in \mathcal{W}, \forall t \in \mathbb{R}_0^+$. We use the same shorthand later for state and input constraints. We denote the solution of (1) with initial state $x(0)$, input $u(\cdot)$, and disturbance $w(\cdot)$ at time t as $\xi(x(0), u(\cdot), w(\cdot), t)$. The measurement of the system is modeled by a function h, returning the measured state $\hat{x}(t)$ subject to a compact set of measurement errors $\mathcal{V} \subset \mathbb{R}^o$:

$$\hat{x}(t) \in \hat{\mathcal{X}}(t) = \{h(x(t), \eta(t)) \,|\, \eta(t) \in \mathcal{V}\}.$$

If not all states are measurable, $\hat{\mathcal{X}}(t)$ can also be obtained by a set-based observer [9], [15].

The goal is to find an MPC controller which steers the system from an initial state $x(0) \in \mathcal{X}$ in finite time into a goal set $\mathcal{X}_f$ while minimizing some cost function. At the same time, the controlled system must satisfy state and input constraints despite disturbances and measurement noise, i.e.,

$$\xi\big(x(0), u(\cdot), w(\cdot), \cdot\big) \in \mathcal{X}, \tag{2}$$

$$u(\cdot) \in \mathcal{U}, \tag{3}$$

where $\mathcal{X}$ and $\mathcal{U}$ are both convex sets in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively.

III. Reachset Model Predictive Control

In this section, we present our novel reachset model predictive control approach. After an overview, we provide required definitions and further detail our approach. In the end, we show all properties in the main theorem.

Overview

The basic idea of our reachset MPC is shown in Fig. 1. Starting from the solution of the previous step (Fig. 1(a)), we obtain a measurement $\hat{x}(t)$ at time t (Fig. 1(b)). As there might be measurement noise, we only know that we are in some uncertain set $\hat{\mathcal{X}}(t)$, which is a singleton when the state can be precisely measured. Based on this measurement, we are looking for the optimal controller which steers the system to the goal set $\mathcal{X}_f$. Since we cannot optimize for an infinite time horizon, we use a dual-mode MPC [20]. This means we consider a final prediction horizon of length t_N and require that the prediction ends in a terminal region Ω (defined formally later in Def. 4), for which we know a safe and stabilizing controller.

Based on the obtained measurement, we optimize a new reference trajectory $x_{ref}(\cdot|t)$, which is tracked with a fixed feedback controller. To solve the optimization problem and to compute the reachable set, we need some time t_c, and we apply the controller from the previous prediction to the system during this time. Using reachability analysis, we predict where we end after the optimization and computation of the reachable set and use this set $\hat{\mathcal{X}}(t+t_c|t)$ as the initial set for our optimization problem (Fig. 1(c)). We use the notation $(t + t_c|t)$ to refer to the prediction for time $t + t_c$ made at time t. For efficiency reasons, we solve the optimization problem for the center trajectory only, but with tightened constraints (Fig. 1(d)). We then use reachability analysis to check if all possible solutions $\hat{\mathcal{X}}(\cdot|t)$ are guaranteed to satisfy all constraints (Fig. 1(e)). Only if this is the case, and if the computations finish in the allocated time t_c, we apply the new, guaranteed-safe solution. If not, we use a feasible solution which consists of the solution from the previous step, extended by the safe controller from the terminal region (Fig. 1(f)). Therefore, under the common assumption that we know a feasible trajectory at the initial time, we always know a feasible solution, which we can use as a backup if we cannot find a better feasible solution in the available time. We then apply the solution for time Δt before we start the next optimization problem based on the new measurement. The feasible solution is defined as:

Definition 1 *The feasible solution is a possible non-optimal input trajectory, which leads to trajectories $\xi\big(x(t), u(\cdot), w(\cdot), \cdot\big)$ satisfying the constraints* (2)-(3) *and ends in the terminal region Ω after time t_N: $\xi\big(x(t), u(\cdot), w(\cdot), t + t_N\big) \in \Omega$.*

After defining reachable sets, we explain all steps of our approach in detail and discuss the guarantees at the end of this section.

Reachability Analysis

To ensure the satisfaction of constraints despite disturbances and measurement noise, we use reachable sets:

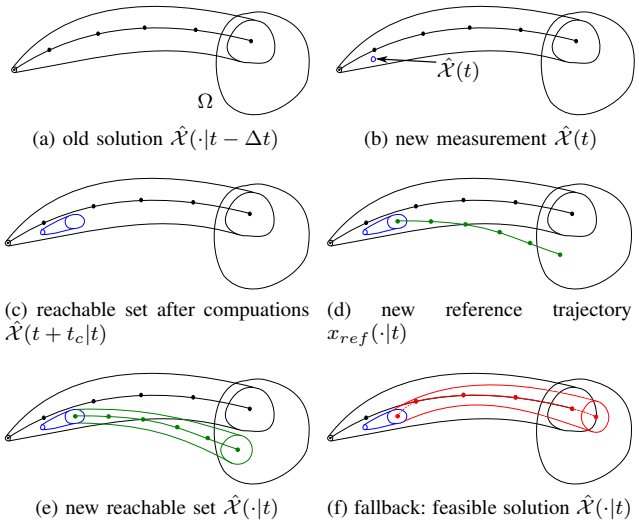
Definition 2 *For a system* (1)*, the reachable set $\mathcal{R}_{t,\mathcal{U},\mathcal{W}}(\mathcal{S}) \subset \mathbb{R}^n$ for a time t, inputs $u(\cdot) \in \mathcal{U} \subset \mathbb{R}^m$, disturbances $w(\cdot) \in \mathcal{W} \subset \mathbb{R}^d$, and a set of initial states $\mathcal{S} \subset \mathbb{R}^n$ is the set of end states of trajectories starting in $\mathcal{S}$ after time t, i.e.,*

$$\mathcal{R}_{t,\mathcal{U},\mathcal{W}}(\mathcal{S}) = \big\{x(t) \in \mathbb{R}^n | \exists x(0) \in \mathcal{S}, u(\cdot) \in \mathcal{U}, w(\cdot) \in \mathcal{W} : \xi\big(x(0), u(\cdot), w(\cdot), t\big) = x(t)\big\}.$$

The reachable set over a time interval $[t_1, t_2]$ is the union of all reachable sets for these time points, i.e.,

$$\mathcal{R}_{[t_1,t_2],\mathcal{U},\mathcal{W}}(\mathcal{S}) = \bigcup_{t \in [t_1,t_2]} \mathcal{R}_{t,\mathcal{U},\mathcal{W}}(\mathcal{S}).$$

If we consider the reachable set for a system with feedback $u_{fb}(\hat{x}(t))$, then we denote by $\mathcal{R}_{t,u_{fb},\mathcal{W}}(\mathcal{S})$ the reachable set



(a) old solution $\hat{\mathcal{X}}(\cdot|t-\Delta t)$ (b) new measurement $\hat{\mathcal{X}}(t)$

(c) reachable set after compuations $\hat{\mathcal{X}}(t+t_c|t)$ (d) new reference trajectory $x_{ref}(\cdot|t)$

(e) new reachable set $\hat{\mathcal{X}}(\cdot|t)$ (f) fallback: feasible solution $\hat{\mathcal{X}}(\cdot|t)$

Fig. 1. Illustration of our reachset MPC approach: Beginning with a feasible solution set $\hat{\mathcal{X}}(\cdot|t-\Delta t)$ from the previous time step (a), we obtain the measurement of the (possibly uncertain) state at time t (b). Based on this set of possible states, we compute the reachable set $\hat{\mathcal{X}}(t+t_c|t)$ (blue) for the time t_c which we need to solve the optimization problem (c). Starting with the center of this reachable set, we optimize the reference trajectory $x_{ref}(\cdot|t)$ (green) for the time horizon t_N (d). After the optimization, we compute the corresponding reachable set $\hat{\mathcal{X}}(\cdot|t)$ (green) (e). If all constraints are satisfied for the reachable set, we use the new reference trajectory and continue with the next iteration at time $t+\Delta t$. If the solution is not feasible or is not computed in time, we follow the feasible solution (red) from the previous time step, which is extended by the auxiliary controller in the terminal region Ω (f).

obtained if we consider the closed-loop dynamics $\dot{x}(t) = f(x(t), u_{fb}(\hat{x}(t)), w(t))$ subject to disturbances and measurement errors. Since it is not possible to compute exact reachable sets for most systems [23], we compute over-approximations instead.

We represents sets by zonotopes due to their favorable properties for reachability analysis [2]:

Definition 3 *A set is called a zonotope if it can be written as*

$$\mathcal{Z} = \left\{ x \in \mathbb{R}^n \middle| x = c + \sum_{i=1}^{p} G\lambda, \lambda \in [-1,1]^p \right\}.$$

Here, $c \in \mathbb{R}^n$ defines the center of the zonotope, and $G \in \mathbb{R}^{n\times p}$ its generator matrix. We use $\langle c, G\rangle$ as a more concise notation of $\mathcal{Z}$.

Dual Mode MPC

As is common in MPC, we use dual mode MPC [20] to limit the prediction horizon. We use the control law

$$u_\Omega(\hat{x}(t)) = K_\Omega \hat{x}(t) \tag{4}$$

to stabilize a terminal region Ω and the control law

$$u_{MPC}(\hat{x}(t)) = v(t) + K(\hat{x}(t) - x_{ref}(t)) \tag{5}$$

which controls the system into the terminal region. Here, $v(t)$ denotes the reference input, which is optimized online, and x_{ref} refers to the corresponding state trajectory. The feedback matrices $K \in \mathbb{R}^{m\times n}$ and $K_\Omega \in \mathbb{R}^{m\times n}$ can be different from each other, and K can even be time-varying as discussed at the end of this section. We use linear controllers for faster computation times; however, all concepts presented also work for nonlinear controllers. The terminal region Ω is defined as a region of attraction in which the state and input constraints are satisfied:

Definition 4 *Given a dynamical system of the form* (1) *and a terminal control law* (4)*. The terminal region Ω, $\mathcal{X}_f \subseteq \Omega \subseteq \mathcal{X}$, is defined as*

$$\Omega = \left\{ x \middle| \forall \eta \in \mathcal{V} : h(x,\eta) \in \bar{\Omega} \right\},$$

with

$$\begin{aligned}\bar{\Omega} = \Big\{ x \Big| \forall t \in \mathbb{R}_0^+, \forall \hat{x}(t) \in \hat{\mathcal{X}}(t), \forall w(t) \in \mathcal{W}, \exists t_f \in \mathbb{R}_0^+ : \\ \xi(x, u_\Omega(\hat{x}(\cdot)), w(\cdot), t_f) \in \mathcal{X}_f, \\ \xi(x, u_\Omega(\hat{x}(\cdot)), w(\cdot), t) \in \mathcal{X}, \\ u_\Omega\big(\xi(x, u_\Omega(\hat{x}(\cdot)), w(\cdot), t)\big) \in \mathcal{U} \Big\}.\end{aligned}$$

Using a terminal region is standard in many MPC approaches and is required to provide guarantees beyond the finite prediction horizon [20]. It is computed before the controller is applied online. There exist different ways to compute an approximation of an invariant set of a controller; many of them use Lyapunov functions, which might be hard to find in practice. While a region of attraction can also be computed using Lyapunov functions, there also exist methods to compute them automatically and in many cases more efficiently using reachable sets [11]. The region of attraction is usually much larger than a positive invariant set, which provides more flexibility to our approach. In addition, by checking the satisfaction of the constraints during the execution of the algorithm from [11], we can automatically compute a safe region of attraction, i.e., a region of attraction for which the state and input constraints are satisfied despite disturbances.

As is common in dual mode MPC, we also use this terminal region to obtain the feasible solution as a backup plan by using the remainder of the previous solution:

$$v_f(\tau|t+\Delta t) = v(\tau|t) \text{ for } \tau = [t+\Delta t + t_c, t+t_N]. \tag{6}$$

Once we reach the terminal region at time $t+t_N$, we switch to the terminal controller (4).

During operation, we compute future reachable sets $\hat{\mathcal{X}}(t+\Delta t|t) = \mathcal{R}_{\Delta t, u_{MPC}, \mathcal{W}}(\hat{\mathcal{X}}(t))$ based on the current input trajectory $v(\cdot|t)$, with $\hat{\mathcal{X}}(t)$ composed of the measured state $\hat{x}(t)$ plus measurement uncertainty. Note that even though $\hat{\mathcal{X}}(t)$ might be partly outside of the reachable set, we know from the over-approximative nature of the reachability analysis that the real state $x(t)$ must lie inside the reachable set from the previous step, i.e., $x(t) \in \hat{\mathcal{X}}(t|t-\Delta t)$. Therefore, we only have to consider the intersection $\hat{\mathcal{X}}(t) \cap \hat{\mathcal{X}}(t|t-\Delta t)$ as the initial set for the next optimization. This is a common approach used in set-based observers [9], [15].

Considering the Computation Time

When starting the optimization for a new measurement, we consider its computation time t_c. To be safe, we need to know the reachable set after t_c due to uncertainties and disturbances:

$$\hat{\mathcal{X}}(t+t_c|t) = \mathcal{R}_{t_c,u_{MPC},\mathcal{W}}(\hat{\mathcal{X}}(t)).$$

By applying the reference trajectory plus feedback controller from the previous optimization, we know that the reachable set after the optimization time is inside the reachable set from the previous optimization.

The allowed computation time t_c for the optimization and reachability analysis is a user-defined design parameter. Note that t_c can be estimated quite well by restricting the iterations of the optimization algorithm and by considering the fact that the computation time for the reachability analysis scales approximately linear with the considered time horizon. However, inappropriate values of t_c do not impede the desired properties in (2)-(3), as we can always go back to the feasible solution if t_c is not sufficient to find a new solution. We compute the reachable set for this alloted time (see Fig. 1(c)). If the optimization algorithm finishes before that, we keep following the previous solution until the designated time, from which point on we apply the new solution. If we reach this point in time without a new feasible solution, we simply keep following the previous feasible solution and start a new optimization (see Fig. 1(f)).

Contraction Constraint

An important consideration in MPC is to ensure the convergence to the goal set in a finite amount of time. While this could be done using Lyapunov functions, we use an approach similar to [6] which does not require a Lyapunov function. Through the construction of the terminal region using the approach from [11], we know that after reaching the terminal region, we converge in finite time to the desired goal set. Therefore, we only have to ensure that we converge in finite time to the terminal region. To do so, we introduce the distance operator from [6]:

Definition 5 *Given sets $\hat{\mathcal{X}}$ and $\Phi = 1/(1+\alpha)\Omega$, with $\alpha \in \mathbb{R}^+$, $\|\hat{\mathcal{X}}\|_\Phi$ is defined as*

$$\|\hat{\mathcal{X}}\|_\Phi = \min \beta, \text{ s.t. } \hat{\mathcal{X}} \subseteq (1+\beta)\Phi, \beta \geq 0.$$

As mentioned in [6], $\|\hat{\mathcal{X}}\|_\Phi$ is equal to zero if and only if $\hat{\mathcal{X}} \subseteq \Phi$, and if $x \notin \Omega$, it follows that $\|x\|_\Phi > \alpha$. The authors also show that if Φ is a polyhedron defined by the intersection of half-spaces of the form $\Phi = \{x : d_i^T x \leq e_i, i \in \{1,\dots,p\}\}$ that contains the origin ($e_i > 0, i \in \{1,\dots,p\}$) and $\hat{\mathcal{X}} = \langle c, G\rangle$ is a zonotope, then $\|\hat{\mathcal{X}}\|_\Phi$ can be obtained from the equality

$$\|\hat{\mathcal{X}}\|_\Phi = \max\left\{0, \max_{i=1,\dots,p} \frac{d_i^T c - e_i + \|G^T d_i\|_1}{e_i}\right\},$$

where $\|G^T d_i\|_1$ denotes the sum of the absolute values of vector $G^T d_i$. By defining the distance with respect to the tighter set Φ, we ensure a desired contraction rate, as shown later in Thm. 1.

Optimal Control Problem

The optimization problem which is solved online at time t is given by

$$\min_{v(\cdot|t)} J\left(\hat{\mathcal{X}}(t+t_c|t), v(\cdot|t)\right) \tag{7}$$
$$= \min_{v(\cdot|t)} \int_{t+t_c}^{t+t_N} L(x_{ref}(\tau|t), v(\tau|t))d\tau + V(x_{ref}(t+t_N|t))$$

s.t.

$$x_{ref}(t+t_c|t) = \text{center}(\hat{\mathcal{X}}(t+t_c|t)), \tag{8}$$
$$\dot{x}_{ref}(t+\tau|t) = f(x_{ref}(t+\tau|t), v(t+\tau|t), 0), \quad \forall \tau \in [t+t_c, t+t_N], \tag{9}$$
$$v(\tau|t) \in \bar{\mathcal{U}}(\tau|t), \quad \forall \tau \in [t+t_c, t+t_N], \tag{10}$$
$$x_{ref}(\tau|t) \in \bar{\mathcal{X}}(\tau|t), \quad \forall \tau \in [t+t_c, t+t_N], \tag{11}$$
$$x_{ref}(t+t_N|t) \in \bar{\Phi}, \tag{12}$$
$$\sum_{k=1}^{\bar{N}(t)-1} \|x_{ref}(t+k\Delta t|t)\|_\Phi - \sum_{k=1}^{\bar{N}(t-\Delta t)-1} \|x_{ref}(t-\Delta t+k\Delta t|t-\Delta t)\|_\Phi < -\bar{\alpha}, \tag{13}$$

where $\text{center}(\hat{\mathcal{X}}(t+t_c|t))$ refers to the center of the zonotope $\hat{\mathcal{X}}(t+t_c|t)$ and $\bar{N}(t) = \min_{k\in\mathbb{N}} x_{ref}(t+k\Delta t|t) \in \bar{\Phi}$.

We minimize the cost function $J(\cdot)$ in (7), consisting of a positive definite state cost $L(\cdot)$ and a positive definite terminal cost $V(\cdot)$, with respect to the center trajectory, which starts from the center of the reachable set (8) after t_c. To ensure the satisfaction of the constraints for the disturbed, closed-loop dynamics, we use tightened time-dependent input (10) and state constraints (11), $\bar{\mathcal{U}}(\cdot)$ and $\bar{\mathcal{X}}(\cdot)$, respectively, as discussed later. As is common in dual-mode MPC, we have a terminal constraint (12), which requires that the center trajectory ends in a tightened terminal region $\bar{\Phi}$. Finally, we have a contraction constraint (13) with parameter $\bar{\alpha}$ (not necessarily equal to α), which ensures convergence to the terminal region Ω.

Tightened Constraints

To be able to apply our MPC approach online, we only optimize the center trajectory without computing the reachable sets during this optimization. While it is possible to optimize over reachable sets [28], this is not possible in real-time for fast systems. The authors of [6] propose optimizing over the reachable sets; however, they do not discuss the computation times and their approach is rather conservative as demonstrated later in Sec. IV. Instead, we optimize only the center trajectory and tighten the constraint sets accordingly, such that state and input constraints are met. At the end of the optimization, we perform a reachability analysis to check if all constraints are actually satisfied. If this is not the case, we always have the feasible solution as a safe fallback. We initially guess the size of the reachable set and the resulting inputs from the controller based on the reachable set from the feasible solution and verify the

solution later. This means we take the size of the reachable set of the feasible solution at the corresponding time step, scaled by a factor $\gamma \in \mathbb{R}^+$, and use this set to tighten the constraints sets. To do this in a set-based fashion, we introduce the Minkowski difference denoted by $\ominus$, i.e., the subtraction of two sets, as the complement of the Minkowski sum: for sets $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^n$ we define

$$\mathcal{X} \oplus \mathcal{Y} = \{x + y | x \in \mathcal{X}, y \in \mathcal{Y}\},$$
$$\mathcal{X} \ominus \mathcal{Y} = \{z \subseteq \mathbb{R}^n | z \oplus \mathcal{Y} \subseteq \mathcal{X}\}.$$

This allows us to write the tightened constraints as

$$\begin{aligned}\bar{\mathcal{X}}(t+\tau) &= \mathcal{X} \ominus \gamma\Big(\hat{\mathcal{X}}(t-\Delta t+\tau|t-\Delta t) \\ &\quad \ominus x_{ref}(t-\Delta t+\tau|t-\Delta t)\Big), \forall \tau \in [t_c, t_N], \\ \bar{\mathcal{U}}(t+\tau) &= \mathcal{U} \ominus K\gamma\Big(\hat{\mathcal{X}}(t-\Delta t+\tau|t-\Delta t) \\ &\quad \ominus x_{ref}(t-\Delta t+\tau|t-\Delta t)\Big), \forall \tau \in [t_c, t_N], \\ \bar{\Phi} &= \Phi \ominus \gamma\Big(\hat{\mathcal{X}}(t-\Delta t+t_N|t-\Delta t) \\ &\quad \ominus x_{ref}(t-\Delta t+t_N|t-\Delta t)\Big).\end{aligned}$$

As the reachable sets might change their size, the constraints become time-dependent. If this guess is too conservative, we only obtain a sub-optimal solution; if it is too optimistic, we have to go back to the feasible solution. In any case, we have a safe solution in the end.

Guarantees Through Reachability Analysis

After obtaining the center trajectory, we use the pre-defined feedback controller to compute the reachable set for the closed-loop dynamics. We start from the reachable set $\hat{\mathcal{X}}(t+t_c|t)$ and compute it for the remaining prediction horizon (see Fig. 1(e)). Afterwards, we check if the reachable set satisfies the state and input constraints at all times, if the final reachable set is completely inside the terminal region, and if the contraction constraint is also satisfied for the reachable sets, i.e., we check if $\forall \tau \in [t_c, t_N]$:

$$\hat{\mathcal{X}}(t+\tau|t) \subseteq \mathcal{X}, \quad (14)$$
$$v(t+\tau|t) \oplus K\left(\hat{\mathcal{X}}(t+\tau|t) \ominus x_{ref}(t+\tau|t)\right) \subseteq \mathcal{U}, \quad (15)$$
$$\hat{\mathcal{X}}(t+t_N|t) \subseteq \Phi, \quad (16)$$
$$\sum_{k=1}^{N(t)-1} \|\hat{\mathcal{X}}(t+k\Delta t|t)\|_\Phi - \sum_{k=1}^{N(t-\Delta t)-1} \|\hat{\mathcal{X}}\big(t+(k-1)\Delta t|t-\Delta t\big)\|_\Phi < -\alpha, \quad (17)$$

where we evaluate the contraction constraint (17) only at finitely many time points to obtain a finite cost and where

$$N(t) = \min_{k \in \mathbb{N}} \hat{\mathcal{X}}(t+k\Delta t|t) \subseteq \Phi. \quad (18)$$

To evaluate if the zonotope $\hat{\mathcal{X}}(t+\tau|t) = \langle c, G\rangle$ satisfies convex state and input constraints of the form $\mathcal{X} = \{x \in \mathbb{R}^n | Cx \leq d\}$, we simply have to check if the following inequality holds:

$$Cc + \sum_{i=1}^{p} |Cg^{(i)}| \leq d, \quad (19)$$

with $g^{(i)}$ denoting the i-th column of $G \in \mathbb{R}^{n \times p}$ and where the absolute value and less or equal operators are both performed element-wise. Using this formula and using the fact that the reachability analysis provides us with reachable sets for time intervals in the form of zonotopes, we can efficiently check if the constraints (14)-(17) are satisfied for the reachable sets at all times. If this is the case, we apply the new control input to the system and start with a new iteration step. If the solution does not satisfy all those constraints or if the computation takes longer than the pre-specified time, we apply the input from the feasible solution instead.

Algorithm 1 Reachset MPC Algorithm

Initialize: $t \leftarrow 0, v(\cdot|-\Delta t) \leftarrow$ initial feasible solution
while $\hat{x}(t) \notin \Omega$ **do**
  $u(\tau) \leftarrow v(\tau|t-\Delta t) + K(\hat{x}(\tau) - x_{ref}(\tau|t-\Delta t)), \tau \in [t, t+t_c]$
  $v_f(\cdot|t) \leftarrow$ feasible solution (6)
  $v^*(\cdot|t) \leftarrow$ solution of optimization problem (7)
  if Optimization problem feasible & solved in time & (14)–(17) satisfied **then** $v(\cdot|t) \leftarrow v^*(\cdot|t)$
  else $v(\cdot|t) \leftarrow v_f(\cdot|t)$
  end if
  $u(\tau) \leftarrow v(\tau|t) + K\big(\hat{x}(\tau) - x_{ref}(\tau|t)\big), \tau \in [t+t_c, t+\Delta t]$
  $t \leftarrow t + \Delta t$
end while
$u(\tau) \leftarrow K_\Omega \hat{x}(\tau), \tau \geq t$

Main Theorem

Theorem 1 *If we know an initial feasible solution at $t = 0$, then Alg. 1 remains feasible for all times and the system robustly converges to the goal set $\mathcal{X}_f$ in finite time. During the whole time, the system satisfies the state and input constraints* (2)-(3) *despite disturbances and uncertain measurements.*

Proof: We have to show three things: (i) The system remains recursively feasible, i.e., in each step we can find a feasible solution, (ii) the system reaches the goal set $\mathcal{X}_f$ in finite time, and (iii) the constraints are satisfied at all times despite disturbances and measurement noise. We keep the proof concise, as many parts follow standard robust MPC techniques, as used in [6].

(i) This can be shown by induction:

Base Case: For t=0, we know a feasible solution by assumption.

Induction Hypothesis: If we know a feasible solution at time t, then we can always get a feasible solution at $t + \Delta t$.

Induction Step: For every step at time $t+\Delta t$, we know from

the over-approximative way of computing the reachable set, that we start inside the reachable set of the previous step, i.e., $\hat{\mathcal{X}}(t+\Delta t) \subseteq \hat{\mathcal{X}}(t+\Delta t|t)$, for which we know the remainder of the solution from the previous step, i.e., $v(t+\tau|t), \forall \tau \in [\Delta t, t_N]$. Since the solution at time t is feasible, it ends in the terminal region, where we know by construction that the terminal controller provides a feasible solution, see Def. 4. Therefore, the previous solution extended by the terminal controller, see (6), is always feasible and can be applied if we do not find a better solution in time.

(ii) The terminal region Ω is computed such that any state inside Ω robustly converges to the goal set $\mathcal{X}_f$ in finite time despite disturbances and sensor noise. Therefore, we only have to ensure reaching the terminal region in finite time. From the contraction constraint (17), we enforce reaching the terminal region in at most $(1/\alpha)\sum_{k=1}^{N} \|\hat{\mathcal{X}}(t+k\Delta t|t)\|_\Phi$ steps. If we find a new solution, we know from (17) that this new solution satisfies the rate of at least $-\alpha$. Let us now show that the feasible solution is also guaranteed to have this convergence rate:

$$\begin{aligned}
&\sum_{k=1}^{N(t)-1} \|\hat{\mathcal{X}}(t+k\Delta t|t-\Delta t)\|_\Phi \\
&- \sum_{k=1}^{N(t-\Delta t)-1} \|\hat{\mathcal{X}}(t+(k-1)\Delta t|t-\Delta t)\|_\Phi \\
&= -\|\hat{\mathcal{X}}(t|t-\Delta t)\|_\Phi < -\alpha,
\end{aligned}$$

where we denote by $\hat{\mathcal{X}}(t+k\Delta t|t-\Delta t)$ the resulting reachable set from the feasible solution $v_f(\cdot|t)$. Since $\hat{\mathcal{X}}(t+(N(t-\Delta t)-1)\Delta t|t-\Delta t) \subseteq \Phi$, we know from (18) that $N(t-\Delta t) = N(t)+1$ and that $\|\hat{\mathcal{X}}(t+(N(t-\Delta t)-1)\Delta t|t-\Delta t)\|_\Phi = 0$. Therefore, the difference is only the cost of $-\|\hat{\mathcal{X}}(t|t-\Delta t)\|_\Phi$. Because $\hat{\mathcal{X}}(t|t-\Delta t) \not\subseteq \Omega$, it follows from Def. 5 that $\|\hat{\mathcal{X}}(t|t-\Delta t)\|_\Phi > \alpha$, and therefore the last inequality holds. As we can always revert to the feasible solution, the convergence in finite time is guaranteed.

(iii) Before we apply the new solution, we check the constraints for the over-approximated reachable set of the disturbed system in (14)-(17). If they are satisfied, then the new solution is safe and can be applied. If they are violated, we apply the safe feasible solution; see (i). ■

Extension

As mentioned before, we cannot guarantee that the solution resulting from the reference trajectory which is computed with the tightened constraints (10)-(13) will satisfy the actual constraints (14)-(17). While we are always safe, this might make our approach unnecessarily conservative. One way to overcome the problem without getting too conservative is to compute several possible solutions in parallel. Using different estimations of reachable sets and inputs applied by the feedback controller results in several optimization problems with different constraints. As they are completely independent, we can utilize modern multi-core processors by solving them and using reachability analysis in parallel and thus choose the best feasible solution.

IV. Numerical Example

To compare our reachset MPC control algorithm with the approach from [6], we use the same nonlinear continuous stirred tank reactor (CSTR) system for our numerical example. The model of the reactor for an exothermic, irreversible reaction A $\rightarrow$ B with constant liquid volume is given by [6]:

$$\begin{aligned}
\frac{d\,C_A}{dt} &= \frac{q}{V}\ (C_{Af} - C_A) - k_0\ \exp\left(-\frac{E}{RT}\right) \cdot C_A + w_1\ , \\
\frac{d\,T}{dt} &= \frac{q}{V}\ (T_f - T) - \frac{\Delta H \cdot k_0}{\rho\ C_p}\ \exp\left(-\frac{E}{RT}\right) \cdot C_A \\
&\quad + \frac{U \cdot A}{V \cdot \rho \cdot C_p}\,(T_c - T) + w_2\ ,
\end{aligned} \tag{20}$$

where C_A is the concentration of A in the reactor, T is the temperature of the reactor and T_c is the coolant stream temperature. The system state is defined as $x = \left[\left(C_A - C_A^0\right), \left(T - T^0\right)\right]^T$, and the system input as $u = T_c - T_c^0$, with the steady state $C_A^0 = 0.5\,mol/l$, $T^0 = 350\,K$, $T_c^0 = 300\,K$. The model parameters can be found in [6].

The set of inputs is $\mathcal{U} = [-20,\ 70]\,K$ and the uncertainty $w = [w_1,\ w_2]^T$ is bounded by $w_1 \in [-0.1, 0.1]\,mol/(l\,min)$ and $w_2 \in [-2, 2]\,K/min$. The example does not consider state constraints and assumes that the state can be precisely measured.

In order to determine a terminal region Ω, we compute an LQR controller for the system linearized at the steady state $x_S = [0,\ 0]^T$, which results in $K_\Omega = [66.65, -4.86]$. We then use the approach from [11] to calculate Ω as explained before. The time step size of $\Delta t = 1.8\,s$ and a prediction horizon of $t_N = 19.8\,s$, which is equal to $N = 11$ time steps, are the same as in [6]. We keep the reference inputs constant in each time step. The cost functions $L(x, v) = v^T R_c\, v$ and $V(x) = x^T Q_c\, x$ are applied; $R_c = 10^{-12}$, and Q_c is a diagonal matrix with 100 and 1 on the diagonal. Since no cost function is provided in [6], we use these parameters to best approximate their trajectory. We use $\alpha = \bar{\alpha} = 0.1$ for the contraction parameter and $\bar{\mathcal{U}} = [-18,\ 68]\,K$ for the tightened input constraints. For the control law $u_{MPC}(x)$ we apply a time-varying feedback matrix K, where at each time step k, we obtain a new K as an LQR controller for the system linearized at $x^* = \left(x_{ref}\,(t+k\Delta t|t) + x_{ref}\,(t+(k+1)\Delta t|t)\,\right)/2$ and with input weighting matrix $R = 100$ and state weighting matrix Q as the identity. In order to reproduce the behavior of the disturbed system during the execution of the algorithm, we simulate the model (20) with random values for the disturbances w. For the allocated optimization time we use the value $t_c = 0.54\,s$.

Our algorithm is implemented in MATLAB and we use the ACADO toolbox [13] to solve the optimal control problems with a multiple shooting algorithm. For the reachable set computation we use the CORA toolbox [3]. All computations are performed on a 2.9GHz quad-core i7 processor with 32GB memory and without using parallel computing.

The initial solution for the first numerical example with initial state $x_0 = [-0.15,\ -45]^T$ is displayed in Fig. 2.

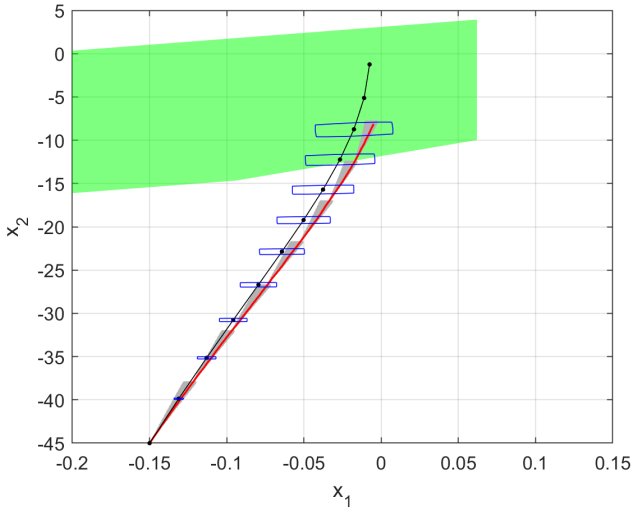


Fig. 2. Center trajectory (black) and reachable sets at discrete time points (blue) of the initial solution for our approach. A resulting trajectory of the real system is shown in red, its reachable set in gray, and the terminal region Ω in green.

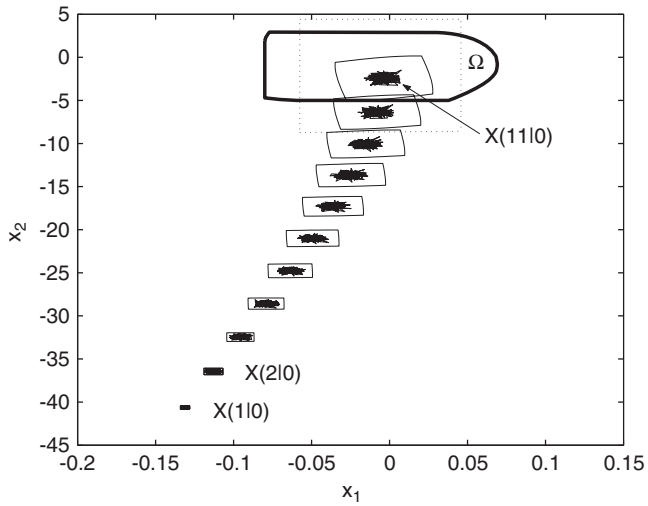


Fig. 3. Initial solution with reachable sets for the rMPC approach, taken from [6].

During Alg. 1, the maximum computation time for the optimization and reachability analysis is $0.51\,s < t_c$, which means that we are able to perform all computations in real time. As a comparison to our algorithm, Fig. 3 shows the initial solution of the robust MPC (rMPC) approach from [6] for the same example. It is clearly visible from Fig. 2 and Fig. 3 that our reachable sets are smaller than the ones computed with rMPC. Small reachable sets are advantageous because they minimize the probability that the input or state constraints are violated. In addition, there is also a better chance that the sets are located inside the terminal region. Furthermore, the rMPC algorithm exhibits several major disadvantages that our approach is able to avoid: First, it does not provide formal safety guarantees for time-continuous systems, as it only considers time-discretized systems. Second, rMPC directly optimizes over the reachable sets, which leads to large computation times, because the reachable sets have to be calculated for each iteration of the optimization algorithm. To avoid this, we only optimize the center trajectory and compute the reachable sets only once after the optimization. Third, the technique that rMPC uses for reachability analysis results in larger over-approximations of the real reachable set of the system, as their technique is more conservative than our approach.

In order to compare our approach with the rMPC algorithm, we use the same parameters and same initial point as the authors in [6]. However, the example is not really suited for a good comparison of control approaches, because to stabilize the system from this initial point, the maximal available control input has to be applied for nearly the whole time horizon. This does not leave much room for the other objectives like minimization of the cost function or counteracting disturbances. Therefore, we provide a second example for the initial point $x_0 = [-0.3, -30]$. Compared to the case above, we changed the final prediction horizon to $t_N = 9\,s$ and the input weighting matrix to $R_c = 0.9$. The results are displayed in Fig. 4. For this example, the maximum computation time for optimization and reachability analysis is $0.37\,s \leq t_c$ This example nicely demonstrates that our repeated optimization enables finding feasible trajectories that have a lower cost than the initial solution.

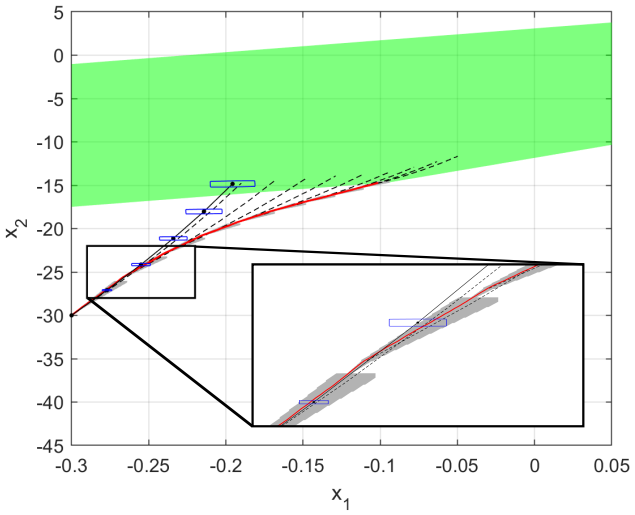


Fig. 4. Our approach for a different initial point with terminal region Ω (green), center trajectory (solid black) and reachable sets at discrete time points (blue) of the initial solution, center trajectories for all iterations (dashed black), real system trajectory (red), and reachable set for the real system trajectory (gray). The resulting reachable sets can be seen better in the magnified section.

V. Discussion of the Algorithm

The computational complexity for our optimization is the same as for regular MPC. During operation, we solve the optimal control problem (8)-(13). Since we solve it only for a single state, we can use the same solvers which are developed for solving nonlinear programs and which are used for existing MPC. The only additional computation effort

is the reachability analysis [2], which has a complexity of $\mathcal{O}(n^3)$, with n denoting the dimension of the state space.

Because this computation only has to be performed once for the whole time horizon, we are able to do these computation in real-time, as shown in the numerical example. Since we do not optimize over the reachable sets and therefore are not able to obtain a global optimal solution (which is not feasible for nonlinear programs in general), we save a lot of computation time while still guaranteeing safety.

An advantage of our approach is that any kind of feedback controller can be used to track the center trajectory and counteract disturbances. It is also not necessary to compute the invariant set or some contraction set which has to hold everywhere in the state space. Instead, we compute the actual reachable set based on the predicted future situation, resulting in a less conservative solution.

VI. Conclusion

We present a novel reachset MPC algorithm which combines reachability analysis with dual-mode MPC. This allows us to prove safety for continuous-time, nonlinear systems with disturbances and uncertain measurements. Due to the online computation of reachable sets, we are not restricted to fixed-size tubes as often seen in literature and therefore are less conservative. In addition, we directly take computation times into account and optimize the trajectory based on the set of initial states after the computation, rather than applying inputs that are computed for states which are measured before the optimization begins.

Compared to the few existing MPC techniques which use reachability analysis, our approach has significant advantages, as we are able to provide guarantees for continuous-time systems and we are able to consider measurement noise and computation times, which are neglected by others. We illustrate the advantages of our approach compared to an existing approach in the numerical example and also show that the computations can be performed in real-time.

The resulting controller has a simple structure, and it can be implemented using standard reachability solvers. In addition, as we do not need to know Lyapunov functions, our approach is easy to use and therefore appealing in practice.

References

[1] A. Alessio and A. Bemporad. *A Survey on Explicit Model Predictive Control*, pages 345–369. Springer, 2009.

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

[3] M. Althoff. An introduction to CORA 2015. In *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 120–151, 2015.

[4] E. Asarin, T. Dang, G. Frehse, A. Girard, C. Le Guernic, and O. Maler. Recent progress in continuous and hybrid reachability analysis. In *Proc. of the IEEE Conference on Computer Aided Control Systems Design*, pages 1582–1587, 2006.

[5] A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, pages 207–226. Springer, 1999.

[6] J. Bravo, T. Alamo, and E. Camacho. Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica*, 42(10):1745 – 1751, 2006.

[7] H. Chen, C. W. Scherer, and F. Allgöwer. A game theoretic approach to nonlinear robust receding horizon control of constrained systems. In *Proc. of the American Control Conference*, pages 3073–3077, 1997.

[8] X. Chen, M. Althoff, and F. Immler. ARCH-COMP17 category report: Continuous systems with nonlinear dynamics. In *ARCH17. 4th International Workshop on Applied Verification of Continuous and Hybrid Systems*, pages 160–169, 2017.

[9] C. Combastel. A state bounding observer based on zonotopes. In *Proc. of the European Control Conference*, pages 2589–2594, 2003.

[10] M. de la Pena, A. Bemporad, and C. Filippi. Robust explicit MPC based on approximate multi-parametric convex programming. In *43rd Conference on Decision and Control*, pages 2491–2496. IEEE, 2004.

[11] A. El-Guindy, D. Han, and M. Althoff. Estimating the region of attraction via forward reachable sets. In *Proc. of the American Control Conference*, pages 1263–1270. IEEE, 2017.

[12] A. Grancharova, T. A. Johansen, and P. Tøndel. *Computational Aspects of Approximate Explicit Nonlinear Model Predictive Control*, pages 181–192. Springer, 2007.

[13] B. Houska, H. Ferreau, and M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[14] W. Langson, I. Chryssochoos, S. Raković, and D. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125 – 133, 2004.

[15] V. T. H. Le, C. Stoica, T. Alamo, E. F. Camacho, and D. Dumur. Zonotopic guaranteed state estimation for uncertain systems. *Automatica*, 49(11):3418–3424, 2013.

[16] D. Limon, T. Alamo, J. M. Bravo, E. F. Camacho, D. R. Ramirez, D. Muñoz de la Peña, I. Alvarado, and M. R. Arahal. *Interval Arithmetic in Robust Nonlinear MPC*, pages 317–326. Springer, 2007.

[17] L. Magni, G. De Nicolao, R. Scattolini, and F. Allgöwer. Robust model predictive control for nonlinear discrete-time systems. *International Journal of Robust and Nonlinear Control*, 13(3-4):229–246, 2003.

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

[19] D. Q. Mayne, M. M. Seron, and S. V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219 – 224, 2005.

[20] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993.

[21] G. Pin, D. M. Raimondo, L. Magni, and T. Parisini. Robust model predictive control of nonlinear systems with bounded and state-dependent uncertainties. *IEEE Transactions on Automatic Control*, 54(7):1681–1687, 2009.

[22] E. N. Pistikopoulos. Perspectives in multiparametric programming and explicit model predictive control. *AIChE Journal*, 55(8):1918–1925, 2009.

[23] A. Platzer and E. M. Clarke. The image computation problem in hybrid systems model checking. In *International Workshop on Hybrid Systems: Computation and Control*, pages 473–486, 2007.

[24] D. Raimondo, S. Riverso, C. Jones, and M. Morari. A robust explicit nonlinear MPC controller with input-to-state stability guarantees. *IFAC Proceedings Volumes*, 44(1):9284 – 9289, 2011.

[25] S. V. Raković, B. Kouvaritakis, M. Cannon, C. Panos, and R. Findeisen. Parameterized tube model predictive control. *IEEE Transactions on Automatic Control*, 57(11):2746–2761, 2012.

[26] S. V. Raković, B. Kouvaritakis, R. Findeisen, and M. Cannon. Homothetic tube model predictive control. *Automatica*, 48(8):1631–1638, 2012.

[27] M. Rubagotti, D. M. Raimondo, A. Ferrara, and L. Magni. Robust model predictive control with integral sliding mode in continuous-time sampled-data nonlinear systems. *IEEE Transactions on Automatic Control*, 56(3):556–570, 2011.

[28] B. Schürmann and M. Althoff. Optimal control of sets of solutions to formally guarantee constraints of disturbed linear systems. In *Proc. of the American Control Conference*, pages 2522–2529, 2017.

[29] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone. Robust online motion planning via contraction theory and convex optimization. In *Proc. of the International Conference on Robotics and Automation*, pages 5883–5890. IEEE, 2017.

[30] V. M. Zavala and L. T. Biegler. The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica*, 45(1):86 – 93, 2009.