

Robust Optimization for MPC

Boris Houska and Mario E. Villanueva

1 Introduction

This chapter aims to give a concise overview of numerical methods and algorithms for implementing robust model predictive control (MPC). In contrast to nominal (certainty-equivalent) MPC, which is by now used for many industrial processes, robust MPC has—at least so far—found much fewer real-world applications. On the one hand, this is due to the fact that nominal MPC often exhibits a certain robustness—as a feedback controller it is inherently able to reject disturbances. Thus, for applications where safety is less critical, a robust MPC formulation, which explicitly models the influence of uncertainty, might simply not be needed. On the other hand, the limited deployment of robust MPC controllers in real-life may very well be due to the numerical challenges associated with their implementation. These challenges range from the need to model the uncertainty affecting the process, to the intractability of general nonlinear formulations—therefore, restricting real-time implementations of robust MPC to simplified models or conservative approximations of the general nonlinear problem. It is also important to notice that the modeling decisions and the problem formulation often influence the choice of an appropriate numerical method. Thus, if one has a process for which it is necessary to explicitly take robustness aspects into account when designing an MPC controller, it is important to know about which tools and algorithms are available.

The focus of the present chapter is to discuss convex approximations of linear robust MPC as well as numerical methods for nonlinear robust MPC, leading to practical implementations. In particular, the advantages and disadvantages of various approaches are explained. Our aim is to explain and summarize highlights of the robust MPC literature from a new and somehow unifying perspective, but, as such,

B. Houska and M.E. Villanueva
School of Information Science and Technology, ShanghaiTech University, China.
e-mail: [borish,meduardov]@shanghaitech.edu.cn

we do not present any new technical contributions. Moreover, different overviews of robust MPC can be found elsewhere in the literature. For example, the book chapter [69, Chapter 3] by J. Rawlings and D. Mayne as well as the plenary article [64] by S.V. Raković discuss a great variety of methods for robust MPC for linear discrete-time systems as well as the trade-off between computational complexity and accuracy. Because our aim is to give a self-consistent overview of numerical methods for robust MPC, some sections of this chapter may overlap with these existing reviews. Nevertheless, this chapter reviews these methods not only from a different, higher-level perspective but also with a much stronger focus on numerical aspects that arise when dealing with nonlinear continuous-time systems, while more theoretical properties, such as closed-loop stability of the controller, are not addressed.

2 Problem formulation

We assume that we have a dynamic process model of the form

$$\forall t \in \mathbb{R}, \quad \dot{x}(t) = f(x(t), u(t), w(t)),$$

which is affected by an unknown but bounded disturbance input. Here, $x : \mathbb{R} \rightarrow \mathbb{R}^{n_x}$ denotes the state, $u : \mathbb{R} \rightarrow \mathbb{R}^{n_u}$ the control input, and $w : \mathbb{R} \rightarrow \mathbb{R}^{n_w}$ the external disturbance or process noise. The function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \rightarrow \mathbb{R}^{n_x}$ is—unless otherwise stated—nonlinear, but assumed to be integrable in all its arguments and Lipschitz continuous in its first argument. It is assumed that the process model is specified together with¹

1. a closed set $\mathbb{X} \subseteq \mathbb{R}^{n_x}$ modeling the state constraints,
2. a compact set $\mathbb{U} \subseteq \mathbb{R}^{n_u}$ modeling the control constraints,
3. and a compact set $\mathbb{W} \subseteq \mathbb{R}^{n_w}$ modeling the disturbance constraints.

Notice that even if the state $x_0 \in \mathbb{R}^{n_x}$ of the system at time t is known, the future state cannot be predicted accurately. This is due to the fact that the function w is unknown to us. However, one important assumption of MPC is that the system state can be measured, i.e., the controller can react to disturbances in closed-loop mode. In the following, the function $\mu : \mathbb{R} \times \mathbb{X} \rightarrow \mathbb{U}$ denotes a feedback law. For a given initial value $x_0 \in \mathbb{R}^{n_x}$, the associate closed loop system can be written in the form

$$\forall t \in [0, T], \quad \dot{x}(t) = f(x(t), \mu(t, x(t)), w(t)), \quad \text{with } x(0) = x_0. \quad (1)$$

Now, a feedback law $\mu : [0, T] \times \mathbb{X} \rightarrow \mathbb{U}$ is called feasible on the time horizon $[0, T]$ and for a given initial value $x_0 \in \mathbb{R}^{n_x}$, if all solutions x of the above closed-loop

¹ the set of compact and convex and compact sets in $\mathbb{R}^n$ are denoted respectively by $\mathbb{K}^n$ and $\mathbb{K}_C^n$.

system satisfy $x(t) \in \mathbb{X}$ for all $t \in [0, T]$ and for all possible uncertainties w , which satisfy $w(t) \in \mathbb{W}$ for all $t \in [0, T]$.

Robust MPC controllers search at every sampling time for feasible feedback laws μ , which are optimal among all feasible feedback laws on a given finite horizon with respect to a given control performance criterion. Putting aside theoretical and numerical difficulties associated to robust MPC, its practical implementation is mostly analogous to certainty-equivalent MPC. This means that an optimization problem is solved starting from a state measurement x_0 (whenever it is available) and shifting the time in a receding-horizon manner. The optimal control input $u(0) = \mu(0, x_0)$ is then sent to the real process. Throughout this chapter we assume that the state measurement is accurate and that the feedback is instantaneous.

The remainder of this section is devoted to different ways of formulating these problems mathematically, as well as their interconnections and particular challenges.

2.1 Inf-sup feedback model predictive control

A mathematical formulation of robust MPC calls for the optimization over feasible feedback laws as defined in the previous section. Let

$$\xi(t, x_0, \mu, w) = x(t)$$

denote the solution (1) as a function of x_0 , μ , and w . An inf-sup feedback model predictive controller is given by

$$\begin{aligned} \inf_{\mu: \mathbb{R} \times \mathbb{X} \rightarrow \mathbb{U}} \sup_{w: \mathbb{R} \rightarrow \mathbb{W}} & \int_0^T l(\xi(t, x_0, \mu, w)) dt + m(\xi(T, x_0, \mu, w)) \\ \text{s.t.} \quad & \sup_{w: \mathbb{R} \rightarrow \mathbb{W}} h_{\mathbb{X}}(\xi(t, x_0, \mu, w)) \leq 0 \quad \text{for all } t \in [0, T], \end{aligned} \quad (2)$$

where the function $h_{\mathbb{X}}: \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is such that $\mathbb{X} = \{x \in \mathbb{R}^{n_x} \mid h_{\mathbb{X}}(x) \leq 0\}$. Here, $l: \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ denotes the stage cost and $m: \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ the terminal cost. Although the stage cost may also depend on the control, i.e., $l(\xi(t, x_0, \mu, w), \mu(t, \xi(t, x_0, \mu, w)))$, this chapter omits this dependency in favor of a more compact notation.

Min-max MPC variants

Mathematical theories, which analyze under which assumptions the above inf-sup formulation can be replaced by a min-max formulation, can mostly be found in the field of functional analysis and optimal control theory, in particular, in the context

of the analysis of viscosity solutions of the Hamilton-Jacobi-Bellman-Isaacs equation [13, 25] as well as Pontryagin's maximum principle [45, 61]. In general, however, it is difficult to check whether a Lebesgue-integrable minimizer μ of (2) exists. In the context of this paper, this question is—to a certain extent—less relevant as long as the objective is bounded from below on the feasible set (this is the case for standard tracking objectives), as numerical methods for solving (2) focus on finding feasible but potentially sub-optimal (also called conservative) points of (2).

2.2 Set-based robust model predictive control

An alternative formulation of the inf-sup feedback MPC uses the concept of reachable sets. Let²

$$X(t, x_0, \mu) = \left\{ x_t \in \mathbb{R}^{n_x} \left| \begin{array}{l} \exists x \in W_{1,2}^{n_x}, \exists w \in L_2^{n_w} : \forall \tau \in [0, t], \\ \dot{x}(\tau) = f(x(\tau), \mu(\tau, x(\tau)), w(\tau)) \\ x(0) = x_0, x(t) = x_t \\ w(\tau) \in \mathbb{W} \end{array} \right. \right\}$$

denote the set of all possible solutions of the closed-loop system at time t for a given feedback law μ and for a given initial value $x_0 \in \mathbb{R}^{n_x}$. Continuous-time robust MPC optimizes the future feedback policy μ by solving the optimization problem

$$\begin{aligned} \inf_{\mu: \mathbb{R} \times \mathbb{X} \rightarrow \mathbb{U}} & \int_0^T \ell(X(t, x_0, \mu)) dt + \mathcal{M}(X(T, x_0, \mu)) \\ \text{s.t.} & \quad X(t, x_0, \mu) \subseteq \mathbb{X} \quad \text{for all } t \in [0, T]. \end{aligned} \quad (3)$$

Here, the function $\ell: \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ denotes a scalar performance measure while the function $\mathcal{M}: \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ denotes the scalar terminal cost.

By definition of the closed-loop reachable set $X(t, x_0, \mu)$, the feasible sets of (2) and (3) coincide. In principle, one could define the objective functions ℓ and $\mathcal{M}$ in dependence on l and m in such a way that complete equivalence of the inf-sup and the set-based formulations is achieved. However, as we will see further below a more important practical consideration is that the choice of the numerical solution method eventually depends on the particular problem formulation and, of course, also on how the objective function is modelled.

² We denote with L_2^n the set of n -dimensional L_2 -integrable functions. Similarly, $W_{1,2}^n$ denotes the associated Sobolev space of weakly differentiable and L_2 -integrable functions on $[0, T]$ with L_2 -integrable weak derivatives.

Topological properties

From a pure mathematical perspective one might argue that problems of the form (3) are rather well-understood in the sense that the topological properties of the sets $X(\cdot, \mu, x_0)$ have been analyzed exhaustively in the context of viability theory [3] and differential inclusions [73]. This connection becomes apparent once the set $X(\cdot, \mu, x_0)$ is interpreted as the solution set of the differential inclusion

$$\dot{x}(t) \in \mathcal{F}(t, x(t), \mu) \quad \text{with} \quad x(0) = x_0$$

with multi-valued right-hand side

$$\mathcal{F}(t, x(t), \mu) = \{f(x(t), \mu(t, x(t)), w(t)) \mid w(t) \in \mathbb{W}\}.$$

For example, it is known that if the right-hand set $\mathcal{F}(t, x(t), \mu)$ of this differential inclusion is convex and compact, the sets $X(t, \mu, x_0)$ are compact under suitable assumptions on μ and f [24]. This result can also be used as a starting point for analyzing whether minimizers of (3) exist [45, 74].

Discrete-time variant

Most of the methods in this chapter can also be applied for the case that the dynamic process is given in the form of a discrete-time system,

$$\forall k \in \{1, \dots, N\}, \quad x_{k+1} = f(x_k, u_k, w_k).$$

Here³, $x = [x_1, \dots, x_N]$, $u = [u_1, \dots, u_N]$, and $w = [w_1, \dots, w_N]$, denote respectively the states, controls and disturbances. The formulation of the associated robust MPC problem is analogous to the continuous-time case,

$$\begin{aligned} \inf_{\mu_0, \dots, \mu_{N-1}: \mathbb{X} \rightarrow \mathbb{U}} \quad & \sum_{k=0}^{N-1} \ell(X_k(x_0, \mu)) + \mathcal{M}(X_N(x_0, \mu)) \\ \text{s.t.} \quad & X_k(x_0, \mu) \subseteq \mathbb{X} \quad \text{for all } k \in \{0, \dots, N\} \end{aligned} \quad (4)$$

with

$$X_{k+1}(x_0, \mu) = \left\{ x_{k+1} \in \mathbb{R}^{n_x} \left| \begin{array}{l} \exists x \in \mathbb{R}^{n_x \times N}, \exists w \in \mathbb{R}^{n_x \times (N-1)} : \forall i \in \{0, \dots, k\}, \\ x_{k+1} = f(x_k, \mu_k(x_k), w_k), \\ w_k \in \mathbb{W} \end{array} \right. \right\}.$$

³ The symbols used for continuous-time models will also be used for discrete-time. Since hybrid models are not considered in this chapter, no confusion should arise from this abuse of notation.

The optimization variables, $\mu_0, \dots, \mu_{N-1} : \mathbb{X} \rightarrow \mathbb{U}$, correspond to the sequence of future feedback policies.

2.3 Numerical challenges

So far, we have introduced two basically equivalent mathematical formulations for the robust optimal feedback control problem. Both formulations showcase different properties of the problem and also motivate different numerical solution methods. Regardless of the chosen formulation, it is clear that robust MPC problems are more difficult to solve than certainty-equivalent MPC problems. The two main reasons are that

1. the predicted vector-valued state trajectory x of certainty-equivalent MPC has to be replaced by a set-valued tube $X(\cdot, x_0, \mu)$, and
2. the optimization variable μ of the robust MPC problem (3) is a feedback law, i.e., a function of the current time t and the current state $x(t)$, rather than a single open-loop control trajectory $u : [0, T] \rightarrow \mathbb{U}$.

Looking at these points, it may appear that (3) involves the extra difficulty of computing the set-valued tube. However, (2) is a bi-level optimization problem, and, as a consequence, the construction of numerical algorithms for computing rigorous solutions will require information about the solution set of the closed-loop system, too.

For anything but the simplest of systems, solving the infinite dimensional robust optimal feedback control problem with high numerical accuracy, may be futile. In part, because the set-valued tube $X(\cdot, \mu, x_0)$ cannot, apart from very simple cases, be stored accurately, and, in part, because algorithms to optimize over general feedback functions do not exist. Therefore, the focus of modern numerical robust MPC algorithms is mostly on computing sub-optimal but feasible feedback laws. Thus, the remainder of this chapter is devoted to presenting numerical algorithms that either exploit particular structures in the problem, e.g. linearity of the dynamics, or construct tractable approximations of the problem.

3 Convex approximations for robust MPC

Because the robust MPC problem needs to be solved in real-time, it would be favorable if we could generate approximate solutions of (3) by solving a convex optimal control problem. Unfortunately, for general robust MPC problems, such convex ap-

proximations are hard to find. Nevertheless, for the case that the system dynamics is affine in the states and controls and under suitable additional conditions on the constraints and objective, it is possible to use tools from the field of convex optimization to construct tractable conservative approximations of (2) or (3). Therefore, the purpose of the following sections is to outline the main successful strategies, which lead to reasonably accurate and scalable convex approximations. Here, “scalable” means that we exclude exhaustive state-space partitioning methods as used in the field of dynamic programming or explicit robust MPC for a moment, which are, however, reviewed at a later point in this chapter.

3.1 Ellipsoidal approximation using LMIs

Let us consider a linear system of the form

$$f(x(t), u(t), w(t)) = A(t)x(t) + B(t)u(t) \quad \text{with} \quad w(t) = [A(t), B(t)] .$$

We assume that the sets $\mathbb{U} = \mathcal{E}(U)$ and $\mathbb{X} = \mathcal{E}(\bar{P})$ are ellipsoids⁴ with given shape matrices⁵ $\bar{P} \in \mathbb{S}_+^{n_x}$ and $U \in \mathbb{S}_+^{n_u}$ and that the uncertainty set⁶

$$\mathbb{W} = \mathbf{co}(\{[A_1, B_1], \dots, [A_m, B_m]\})$$

is a polytope with given vertices $[A_i, B_i] \in \mathbb{R}^{n_x \times (n_x + n_u)}$. Following a (conservative) linear parameterization of the feedback law, $u(t) = K(t)x(t)$, the associated closed-loop system is an uncertain linear system of the form

$$\dot{x}(t) = (A(t) + B(t)K(t))x(t) \quad \text{with} \quad x(0) = x_0 .$$

It can be checked easily that the reachable set of this differential equation can be overestimated by an ellipsoidal tube, $X(t, \mu, x_0) \subseteq \mathcal{E}(P(t))$, if the shape matrix P satisfies the Lyapunov differential inequality

$$\dot{P}(t) \succeq (A(t) + B(t)K(t))P(t) + P(t)(A(t) + B(t)K(t))^T \quad (5)$$

$$P(0) \succeq x_0 x_0^T . \quad (6)$$

This matrix inequality needs to hold for all $t \in [0, T]$ and all matrix-valued functions A, B , which satisfy $[A(t), B(t)] \in \mathbb{W}$. The control can be written as

$$\forall \xi' \in \mathcal{E}(P(t)), \quad K(t)\xi' \in \mathcal{E}(U) \quad \Leftrightarrow \quad K(t)P(t)K(t)^T \preceq U .$$

⁴ In this chapter, $\mathcal{E}(Q) := \{Q^{\frac{1}{2}}v \mid v^T v \leq 1\}$ denotes an n -dimensional ellipsoid with positive semidefinite shape matrix Q .

⁵ The set of symmetric positive semidefinite matrices in $\mathbb{R}^{n \times n}$ is denoted by $\mathbb{S}_+^n$.

⁶ We use the notation $\mathbf{co}(S)$ for the convex hull of a set S .

In order to proceed, one needs to apply two convex analysis “tricks”:

1. We introduce the variable substitution $Y(t) = K(t)P(t)$ to get rid of the bilinear terms in (5). As long as the search is constrained to positive functions $P(t) \succ 0$, this substitution is invertible, i.e., optimizing over the function $Y : [0, T] \in \mathbb{R}^{n_u \times n_x}$ is equivalent to optimizing over the feedback gain matrix $K(t) = Y(t)P(t)^{-1}$.
2. Since the right hand expression in (5) is affine in $w(t) = [A(t), B(t)]$ it is sufficient to enforce this inequality at the vertices of the polytope $\mathbb{W}$ rather than for all points inside this polytope.

In summary, (5) holds for all A, B with $[A(t), B(t)] \in \mathbb{W}$ if

$$\begin{aligned} \dot{P}(t) &\succeq A_i P(t) + P(t) A_i^\top + B_i Y(t) + Y(t)^\top B_i^\top, \quad \forall t \in [0, T] \\ P(0) &\succeq x_0 x_0^\top, \end{aligned}$$

holds for each $i \in \{1, \dots, m\}$, with $Y(t) = K(t)P(t)$. The state and control constraints can now be enforced as

$$P(t) \preceq \bar{P} \quad \text{and} \quad K(t)P(t)K(t)^\top = Y(t)P(t)^{-1}Y(t)^\top \preceq U.$$

The latter inequality is “quadratic-over-linear” in (Y, P) and thus convex. By using Schur complements, this inequality can alternatively be written in the form of the linear matrix inequality

$$\begin{pmatrix} U & Y(t) \\ Y(t)^\top & P(t) \end{pmatrix} \succeq 0,$$

which has the additional advantage that the inverse of $P(t)$ is not needed. Now, a conservative approximation of (3) is given by

$$\begin{aligned} \inf_{P, Y} \int_0^T \ell(\mathcal{E}(P(t))) dt + \mathcal{M}(\mathcal{E}(P(T))) \\ \text{s.t.} \quad \begin{cases} \forall t \in [0, T], \forall i \in \{1, \dots, m\} : \\ \dot{P}(t) \succeq A_i P(t) + P(t) A_i^\top + B_i Y(t) + Y(t)^\top B_i^\top \\ P(0) \succeq x_0 x_0^\top \\ 0 \preceq \begin{pmatrix} U & Y(t) \\ Y(t)^\top & P(t) \end{pmatrix} \\ 0 \prec P(t) \preceq \bar{P}. \end{cases} \end{aligned} \quad (7)$$

If one models the objective in such a way that the expression in the Lagrange and Mayer term are convex in P , for example, with

$$\ell(X') = \max_{\xi' \in X'} \|\xi'\|_2^2 \implies \ell(\mathcal{E}(P(t))) = \lambda_{\max}(P(t)) \quad \text{and} \quad \mathcal{M}(X') = 0,$$

the optimization problem (7) is a convex optimal control problem. Every feasible solution (Y, P) of (7) yields a control law,

$$\mu(t, \xi') = Y(t)P(t)^{-1}\xi',$$

which is a feasible point of the original robust MPC problem (3).

The above linear matrix inequality (LMI) based approximation of robust MPC has (in a very similar variant) been proposed in [39, 82]. Of course, one could think of many other variants and extensions of the convex robust MPC approximation (7). For example, one could try to extend the above analysis by working with affine rather than linear control parameterizations or by extending the formulation for other types of objectives or other uncertainty models. Some of these variants lead to convex optimization problems or at least to optimization problems, which are convex with respect to most of their optimization variables, as discussed extensively in J. Löfberg's Ph.D. thesis [47]. However, the main idea of most of these LMI relaxations is to substitute the equation

$$Y(t) = K(t)P(t)$$

(or similar variable transformations) at some point in the derivation in order to eliminate bilinear terms. For a more general overview on LMIs in systems and control, we refer to the textbook [12], because many of the methods in this book can be used as a starting point to construct LMI relaxations or other types of convex approximations of robust MPC.

Feasibility

In general, (7) is a conservative approximation of the original robust MPC problem. Thus, the semi-definite state constraint, $P(t) \preceq \bar{P}$, may lead to infeasibility—even if the original robust MPC problem was perfectly well-formulated and feasible. Nevertheless, under the additional assumption that there exists a $\bar{Y} \in \mathbb{R}^{n_u \times n_x}$ such that $\bar{P}$ satisfies

$$\forall i \in \{1, \dots, m\}, \quad \begin{cases} 0 \succeq A_i \bar{P} + \bar{P} A_i^\top + B_i \bar{Y} + \bar{Y}^\top B_i^\top \\ \bar{P} \preceq \begin{pmatrix} U & \bar{Y} \\ \bar{Y}^\top & \bar{P} \end{pmatrix}, \end{cases} \quad (8)$$

then $\mathcal{E}(\bar{P})$ is a robust forward invariant set, i.e., (7) is feasible (and remains recursively feasible) as long as the initial value satisfies $x_0 \in \mathcal{E}(\bar{P})$.

3.2 Affine disturbance feedback

A second approach for approximating the robust feedback policy optimization problem by a convex optimization problem is based on affine disturbance feedback parameterization. This approach can be applied to linear systems of the form

$$f(x(t), u(t), w(t)) = Ax(t) + Bu(t) + w(t) .$$

In contrast to the model from the previous section, the matrices $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$ are assumed to be given, i.e., the uncertainty w enters in the form of an additive offset only. Next, the main idea is to introduce a linear feedback parameterization *with memory*, i.e., a control law of the form

$$u(t) = \int_0^t L(t, \tau) x(\tau) d\tau ,$$

where the function $L : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^{n_u \times n_x}$ becomes the new optimization variable. At this point, one should be clear in mind that the formulation of (3) was based on the assumption that the feedback law, $\mu(t, x(t))$, depends on the current state only, but does not have memory. This assumption is not restrictive, as the current (exact!) state measurement contains all the relevant information that is needed to predict the future evolution of the system. However, if we restrict ourselves to affine state feedback laws, it may be that the optimal feedback law depends on previous state measurements, i.e., the principle of information separation into future and past is violated. Now, the main observation is that optimizing over the set of affine state feedback laws with memory is equivalent to optimizing over the class of affine disturbance feedback laws of the form⁷

$$u(t) = \int_0^t M(t, \tau) w(\tau) d\tau ,$$

where $M : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^{n_u \times n_x}$ is the new optimization variable. This equivalence becomes apparent⁸ by noticing that the disturbance function can be computed from $w(\tau) = \dot{x}(\tau) - Ax(\tau) - Bu(\tau)$, if we know the functions x and u on the horizon $\tau \in [0, t]$. The mathematical advantage of this change of variables is that the state at time t ,

$$x(t) = H(t, x_0) \circ M$$

is an affine functional in M . Here, $H(t, x_0)$ denotes an affine operator,

$$H(t, x_0) \circ M = e^{At} x_0 + \int_0^t e^{A(t-\tau)} \left(B \int_0^\tau M(\tau, \tau') w(\tau') d\tau' + w(\tau) \right) d\tau ,$$

⁷ The idea to use affine disturbance feedback parameterization in order to approximate robust MPC with (in different variants) be found in [7, 19] as well as in an early article by J. Löfberg [49].

⁸ Details about this equivalence statement and a more formal proof in the discrete-time setting can be found in an article by Goulart and Kerrigan [28].

mapping the function M to $x(t)$. Consequently, if the sets $\mathbb{X}$ and $\mathbb{U}$ are convex, the set of feasible disturbance feedback functions,

$$\mathcal{C}_M = \left\{ M : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^{n_u \times n_x} \left| \begin{array}{l} \forall w : [0, T] \rightarrow \mathbb{W}, \forall t \in [0, T], \\ \int_0^t M(t, \tau) w(\tau) d\tau \in \mathbb{U} \\ H(t, x_0) \circ M \in \mathbb{X} \end{array} \right. \right\},$$

is convex. Similarly, if we have worst-case objectives of the form

$$\ell(X') = \max_{\xi' \in X'} l(\xi') \quad \text{and} \quad \mathcal{M}(X') = \max_{\xi' \in X'} m(\xi')$$

with convex functions l and m , the corresponding parametric functionals

$$\ell(X(t, \mu, x_0)) = \tilde{\ell}(t, M, x_0) = \max_{w, w(\tau) \in \mathbb{W}} l(H(t, x_0) \circ M)$$

$$\text{and } \mathcal{M}(X(T, \mu, x_0)) = \tilde{\mathcal{M}}(M, x_0) = \max_{w, w(\tau) \in \mathbb{W}} m(H(T, x_0) \circ M)$$

are convex in M , since the maximum over convex functions remains convex. Consequently, a conservative approximation of (3) is given by

$$\inf_{M \in \mathcal{C}_M} \int_0^T \tilde{\ell}(t, M, x_0) dt + \tilde{\mathcal{M}}(M, x_0). \quad (9)$$

This is a convex optimization problem. Of course, in order to solve (9) one still needs to discretize the function M —preferably without destroying convexity. For example, if one uses piecewise constant discretization of M on the 2-dimensional discrete-time grid,

$$\{t_0, t_1, \dots, t_N\} \times \{t_0, t_1, \dots, t_N\}, \quad 0 = t_0 < t_1 < \dots < t_N = T,$$

we end up with a convex optimization problem with $\mathbf{O}(N^2)$ matrix-valued optimization variables, as worked out in an article by Goulart and Kerrigan [28]. Thus, in general, affine disturbance feedback parameterization based robust MPC typically lead to algorithms which scale (at least) quadratically with the discrete-time prediction horizon length N . There are, however, variants which enforce additional structure of the function M , e.g., by setting $M(t, \tau) = 0$ for $|t - \tau| \geq \bar{\tau}$ for a given constant $\bar{\tau} > 0$, which leads to computationally less demanding but more conservative convex approximations of the original robust MPC problem. Methods for analyzing the conservatism of affine disturbance feedback parameterizations in the context of robust MPC can be found in [77].

Moreover, the above linear feedback parameterization with memory is at most as but in general less conservative than the linear feedback parameterization from Section 3.1 as used in the early article by M.V. Kothare and co-workers [39]. A discus-

sion of more general feedback parameterization structures can also be found in the overview article by S.V. Raković [64].

4 Generic methods for robust MPC

There are three main classes of generic numerical algorithms for robust nonlinear MPC, namely, *(Approximate) Dynamic Programming* based approaches, *Scenario-Tree MPC* based approaches, and *Tube-MPC* based approaches.

4.1 Inf-sup dynamic programming

Dynamic programming based methods for robust MPC are, at least in principle, analogous to dynamic programming methods for nominal MPC. This means that one introduces the so-called cost-to-go (or value) function

$$V(t, y) = \inf_{\mu: \mathbb{R} \times \mathbb{X} \rightarrow \mathbb{U}} \sup_{w: \mathbb{R} \rightarrow \mathbb{W}} \int_0^{T-t} l_{\mathbb{X}}(\xi(\tau, y, \mu, w)) d\tau + m(\xi(T-t, y, \mu, w))$$

for $(t, y) \in [0, T] \times \mathbb{R}^{n_x}$, where we use the shorthand

$$l_{\mathbb{X}}(\xi) = \begin{cases} l(\xi) & \text{if } \xi \in \mathbb{X} \\ \infty & \text{otherwise.} \end{cases}$$

It is known since a long time that the function V can formally be obtained as the viscosity solution of an inf-sup Hamilton-Jacobi-Bellman equation (also known under the name “Hamilton-Jacobi-Bellman-Isaacs equation”), which is a partial differential equation (PDE) of the form

$$\begin{aligned} -\frac{\partial}{\partial t} V(t, y) &= \inf_{v \in \mathbb{U}} \sup_{\omega \in \mathbb{W}} \{l_{\mathbb{X}}(y) + \nabla_y V(t, y)^T f(y, v, \omega)\} \\ V(T, y) &= m(y) \end{aligned} \tag{10}$$

on the domain $[0, T] \times \mathbb{R}^{n_x}$. Now, one can use numerical tools from the field of partial differential equations in order to find an approximate solution. If a minimizer exists, an optimal feedback law can be picked as

$$\mu^*(t, y) \in \operatorname{argmin}_{v \in \mathbb{U}} \sup_{\omega \in \mathbb{W}} \{l_{\mathbb{X}}(y) + \nabla_y V(t, y)^T f(y, v, \omega)\}.$$

PDE solvers for Hamilton-Jacobi-Bellman equations have been developed in [56] and there exists efficient software for solving this type of PDEs, e.g., the level-set toolbox by Mitchell and co-workers [57, 55]. An appealing feature of the above Hamilton-Jacobi-Bellman equation is that if we manage to solve this PDE, we can have access to a globally optimal feedback law. This is important for certain non-convex robust MPC scenarios, e.g., in the context of obstacle avoidance problems in robotics [27].

A common criticism of (numerically accurate) generic methods based on a direct solution of (10) is that these approaches can only be applied to problems with a small number of states. This is due to the fact that, at least in general, one has to use a grid in the state space in order to construct accurate approximations of the function V , as discussed in [31]. In practice, the corresponding adaptive grid based methods often only work well for generic problem with $n_x \leq 3$ states. Nevertheless, for the case that the differential equations have additional structure, it is possible to solve (10) for higher dimensional problems. For example, in recent articles by M. Chen, S. Bansal, and co-workers [4, 17] min-max Hamilton-Jacobi Bellman equations have been solved numerically for (simplified) quadcopter models with up to $n_x = 10$ differential states by exploiting the rather particular structures of these models.

Remark: Notice that the inf-sup Hamilton-Jacobi PDE is closely related to its associated sup-inf version, which is obtained by swapping the inf and the sup operation in (10). As discussed in [13], the solutions of the inf-sup and the sup-inf Hamilton-Jacobi PDE coincide under mild technical assumptions. This result can also be expected intuitively, as the feedback is instantaneous. To see this, consider a differential game where both us as well as our adverse player (nature) are choosing all control and disturbance reactions instantaneously. Clearly, if both players can choose their actions continuously in time, one cannot distinguish between who plays first and who plays last. This is in contrast to the discrete-time min-max dynamic programming recursion, which is reviewed below and for which the min and the max operation cannot be exchanged.

Discrete-time variant

The discrete-time analog of the Hamilton-Jacobi-Bellman PDE (10) is known under the name *dynamic programming recursion*,

$$\begin{aligned} V_k(y) &= \min_{v \in \mathbb{U}} \max_{\omega \in \mathbb{W}} l_{\mathbb{X}}(y) + V_{k+1}(f(y, v, \omega)) \\ V_N(y) &= m(y) \end{aligned} \tag{11}$$

for all $k \in \{0, \dots, N-1\}$ and all $y \in \mathbb{R}^{n_x}$. Here, the function sequence $V_N, V_{N-1}, \dots, V_0$ can be found by solving the above backward recursion. Next, an optimal solution of the associated discrete-time robust MPC problem can be found as

$$\mu_k^*(y) \in \operatorname{argmin}_{v \in \mathbb{U}} \max_{\omega \in \mathbb{W}} l_{\mathbb{X}}(y) + V_{k+1}(f(y, v, \omega)) .$$

Discrete-time dynamic programming recursions are the basis for a number of existing robust MPC tools.

1. For the special case that the discrete-time system f is affine, m piecewise convex-quadratic, $\mathbb{U}$, $\mathbb{X}$, and $\mathbb{W}$ polytopic, it can be shown that the functions V_k are piecewise quadratic and can be constructed explicitly [6]. A corresponding robust MPC tool is available as part of the multi-parametric toolbox MPT [32], which can be used in combination with Yalmip [48].
2. For the special case that the discrete-time system f is affine, m piecewise affine, $\mathbb{U}$, $\mathbb{X}$, and $\mathbb{W}$ polytopic, approximate robust dynamic programming methods have been developed in [9, 20], which construct piecewise affine upper- and lower bounds on the function V_k , which leads to a sub-optimal robust MPC controller with guarantees.

Other dynamic programming or approximate dynamic programming tools and methods can be found in [8, 83].

4.2 Scenario-tree MPC

The scenario-tree approach [72], sometimes also referred to as Multi-Stage MPC, is a method for generating optimistic approximations of the discrete-time robust MPC problem (4). This means that if we have a continuous-time problem, we have to discretize this problem first. Now, the main idea is to choose a discrete inner approximation $\underline{\mathbb{W}} = \{\tilde{w}_1, \dots, \tilde{w}_m\} \subseteq \mathbb{W}$ of the disturbance set. We also define the index set $\mathcal{J} = \{1, \dots, m\}$. Next, an optimistic scenario based approximation of (4) can be written in the form

$$\begin{aligned} \inf_{\tilde{x}, \tilde{u}} \quad & \sum_{k=0}^{N-1} \ell(\{\tilde{x}_{k,i_1,\dots,i_k} \mid i_1, \dots, i_k \in \mathcal{J}\}) + \mathcal{M}(\{\tilde{x}_{N,i_1,\dots,i_N} \mid i_1, \dots, i_N \in \mathcal{J}\}) \\ \text{s.t.} \quad & \begin{cases} \forall k \in \{0, \dots, N\}, \forall i_k \in \mathcal{J}, \\ \tilde{x}_{k+1,i_1,\dots,i_{k+1}} = f(\tilde{x}_{k,i_1,\dots,i_k}, \tilde{u}_{k,i_1,\dots,i_k}, \tilde{w}_{i_{k+1}}) \in \mathbb{X} \\ \tilde{u}_{k,i_1,\dots,i_k} \in \mathbb{U} \\ \tilde{x}_0 = x_0 . \end{cases} \end{aligned} \tag{12}$$

The optimization variables of this problem, $\tilde{x} \in \mathbb{R}^{n_x \cdot d_N}$ and $\tilde{u} \in \mathbb{R}^{n_u \cdot d_N - 1}$, contain the states and associated optimal control reactions of all possible scenarios. Notice that one can pick m possible uncertainty $\tilde{w}_{i_1} \in \mathbb{W}$ in the first step. For each of these scenarios, one can choose a control reaction $\tilde{u}_{1,i_1}$. In the second step, there are already m^2 possible scenarios; and so on. Thus, in total, there are $d_N - 1$ possible scenarios with

$$d_N = 1 + m + m^2 + \dots + m^N = \frac{m^{N+1} - 1}{m - 1}.$$

A visualization of this approach for $m = 2$ can be found in Figure 1. Notice that this approach is not rigorous in the sense that it does not necessarily lead to a feasible control law. This is due to the fact that only a finite number of uncertainty scenarios is taken into account. Nevertheless, this optimistic approximation may be sufficiently accurate for practical purposes if m is large. Moreover, if a probability distribution of the disturbance inputs w_k is available, one may choose the points $\tilde{w}_i$ according to this distribution such that additional information about the probability distribution of future states can be inferred.

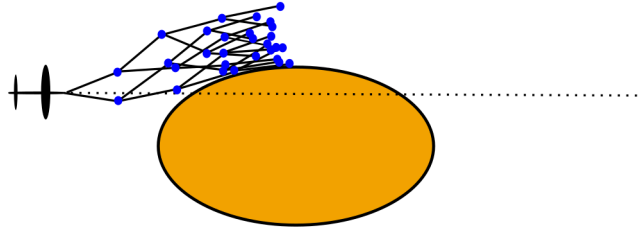


Fig. 1 Visualization of optimized discrete-time scenarios obtained for an obstacle avoidance problem and $m = 2$. After the first discrete-time step, the airplane may be at one out of 2 possible positions in the state-space. After two time steps, there are already 4 possible position, and so on, leading to an exponentially large number of scenarios for long horizons. The corresponding feedback control inputs are optimized in such way that the possible positions of the airplane (dots) never overlaps with the obstacle.

The practical applicability of the scenario-tree approach is limited by the fact that the number of scenarios increases exponentially with the time horizon N . This means that the scenario approach can only be used for small N (in practice often $N \approx 3$). On the other hand, this approach is applicable to robust MPC problems with many states, $n_x \gg 1$, which can be seen as an advantage compared to dynamic programming or tube based approaches. Current research on scenario-tree approaches mostly focus on the development of solvers, which can exploit the particular structure of the optimization problem (12), as well as on heuristics for reducing the number of scenarios [23, 50].

4.3 Tube MPC

As mentioned in the introduction, often the problem formulation motivates the choice for an appropriate numerical method. A detailed analysis of (3) suggests that a direct method for solving this robust MPC problem can be obtained by constructing parametric outer approximations of the tube $X(\cdot, \mu, x_0)$. The main idea (sketched in Figure 2) is simple but elegant: tube-based MPC approaches, as formalized by Raković, Mayne and collaborators [63, 43], compute an outer approximation of the set of all possible states that can be reached under all possible (continuous-time) disturbance scenarios.

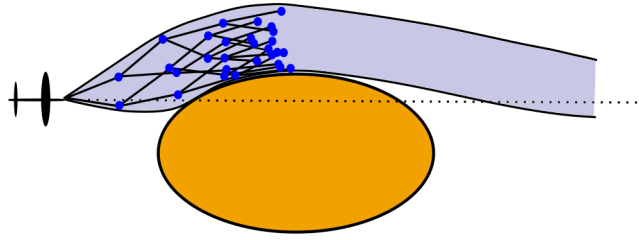


Fig. 2 In contrast to Scenario-Tree MPC from Figure 1, Tube MPC optimizes a single set-valued tube in the state space, which encloses all possible scenarios. The tube (shaded area) may not intersect with the infeasible region in the state-space, in this case an obstacle.

Obviating for a moment that the outer optimization in (3) is still an optimal control problem over feedback laws, the development of scalable numerical algorithms for tube MPC relies on our ability to construct tractable representations of the tube. In particular, the over approximations of the closed-loop reachable sets must be constructed, stored and propagated efficiently.

Due to its importance among robust MPC methods, the next section is devoted to presenting the main strategies used to construct tube MPC based control algorithms.

5 Numerical methods for Tube MPC

In this section we present strategies to construct tube model predictive control algorithms. These strategies present some answers to the two main difficulties of tube MPC: the outer optimization over feedback laws and the practical construction of

tubes for the inner optimization. Both problems can be addressed by appropriate parameterizations, first, of the feedback law, and second, of the reachable set outer approximations.

5.1 Feedback parameterization

The most common class of feedback parameterization based approximations of (3) can be obtained by using affine feedback laws of the form

$$\tilde{\mu}[K, k](t, x) = K(t)x + k(t) .$$

If we substitute $\mu = \tilde{\mu}[K, k]$ in (3), the new optimization variables are the matrix-valued function $K : \mathbb{R} \rightarrow \mathbb{R}^{n_u} \times \mathbb{R}^{n_x}$ and the vector-valued additive offset function $k : \mathbb{R} \rightarrow \mathbb{R}^{n_u}$. In this case, the control constraint,

$$\forall t \in [0, T], \quad K(t)X(t, \tilde{\mu}[K, k], x_0) + k(t) \subseteq \mathbb{U} ,$$

has to be added explicitly in order to ensure that the control law is feasible. Clearly, this and other feedback parameterizations lead to conservative approximations of (3). This means that if the corresponding optimization problem has a solution, we find a control law that ensures that all constraints are satisfied for all possible uncertainty scenarios. In some articles, e.g. in [84], it has been suggested to pre-compute a suitable feedback gain K and only optimize k online, which leads to a simpler optimization problem but an even more conservative approximation of (3). Similarly, one can use more expensive but less conservative feedback parameterizations, e.g., by optimizing over polynomial feedback laws or piecewise affine control laws [6].

The optimization variable μ of the robust MPC controller (3) is sometimes called an “ancillary control law” (see, e.g., [69]), because it is only needed to compute feasible set-valued tubes in the state-space. Thus, even if we substitute a linear feedback parameterization of μ , the actual robust MPC controller realizes a nonlinear control law, because μ is recomputed whenever a new measurement becomes available. However, there are two notable practical variants of robust MPC, which use the feedback law μ explicitly:

1. *Offline Robust MPC*, also called (approximate) *Explicit MPC*, solves (3) approximately offline, e.g., on a sufficiently long horizon (or with periodic boundary conditions on the tube) and over a suitable class of parametric feedback laws, and then uses the optimal feedback law to control the process.
2. *Real-Time Robust MPC* uses an optimized feedback law μ to control the process at a high sampling rate, but updates μ every now-and-then depending on how long the numerical routine needs to solve (3). For this variant, one would usually

optimize over a class of feedback laws that can be evaluated efficiently in online mode.

5.2 Affine set-parameterizations

The formulation of (3) is based on the introduction of the set-valued tubes $X(\cdot, \mu, x_0)$. Unfortunately, it is not possible to store general sets in a computer and, consequently, many set-computation based numerical methods for robust MPC focus on the construction of computer representable sets, which approximate the exact reachable set from outside. Examples for frequently used computer-representable sets are intervals, zonotopes, polytopes, ellipsoids, or polynomial sets. All these sets have in common that they can be written in the form

$$C * \mathbb{E} + c = \{Cx + c \mid x \in \mathbb{E}\},$$

where $\mathbb{E} \subseteq \mathbb{R}^m$ is a compact basis set. The matrix $C \in \mathbb{R}^{n \times m}$ and the vector $c \in \mathbb{R}^n$ are the coefficients of the set parameterization. As the coefficient matrix $[C, c]$ is finite dimensional it can be stored in a computer. For example, in order to store an interval, one would use the unit box,

$$\mathbb{E} = \{x \in \mathbb{R}^n \mid \|x\|_\infty \leq 1\},$$

as the basis set. The coefficient c can in this case be interpreted as the center of the interval box. The matrix C is then required to be non-negative and diagonal such that the diagonal entries can be interpreted as the widths of the box in different coordinate aligned directions. If one allows more general coefficient matrices C , one can also represent rotated (non-coordinate aligned) interval boxes. Table 1 lists a number of affine set parameterizations [16], which are frequently used in the context of robust MPC. Notice that some numerical set-based computing algorithms use variants of the above affine set parameterizations. For example, if we work with ellipsoids with center $c = q$ and symmetric and positive semi-definite (p.s.d.) shape matrix Q ,

$$\left\{ Q^{\frac{1}{2}}x + q \mid \|x\|_2 \leq 1 \right\},$$

it is sometimes more convenient to store the shape matrix Q directly instead of its symmetric square-root $C = Q^{\frac{1}{2}}$.

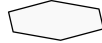
Set	Basis set $\mathbb{E}$	Coefficients	Storage complexity
Interval 	$\{x \in \mathbb{R}^n \mid \ x\ _\infty \leq 1\}$	$C \in \mathbb{R}_+^{n \times n}, c \in \mathbb{R}^n$ C diagonal	$\mathbf{O}(n)$
Zonotope 	$\{x \in \mathbb{R}^m \mid \ x\ _\infty \leq 1\}$	$C \in \mathbb{R}^{n \times m}, c \in \mathbb{R}^n$	$\mathbf{O}(nm)$
Polytope 	$\{x \in \mathbb{R}_+^m \mid \sum_i x_i = 1\}$	$C \in \mathbb{R}^{n \times m}, c \in \mathbb{R}^n$	$\mathbf{O}(nm)$
Ellipsoid 	$\{x \in \mathbb{R}^n \mid \ x\ _2 \leq 1\}$	$C \in \mathbb{R}^{n \times n}, c \in \mathbb{R}^n$ C symmetric and p.s.d.	$\mathbf{O}(n^2)$
Polynomial set 	$\{(1, x_1, x_1 x_2, \dots, x_r^s)^\top$ with $x \in [-1, 1]^r\}$	$C \in \mathbb{R}^{n \times \binom{r+s}{r}}, c = 0$	$\mathbf{O}(nr^s)$

Table 1 List of frequently used computer-representable sets with polynomial storage complexity.

5.3 Tube MPC parameterization

In the context of tube MPC, the key for developing practical implementations is to construct parametric coefficient functions

$$C(\cdot, \mu, x_0) : [0, T] \rightarrow \mathbb{R}^{n_x \times m} \quad \text{and} \quad c(\cdot, \mu, x_0) : [0, T] \rightarrow \mathbb{R}^{n_x},$$

such that

$$\forall t \in [0, T], \quad X(t, \mu, x_0) \subseteq C(t, \mu, x_0) \cdot \mathbb{E} + c(t, \mu, x_0)$$

for a suitable basis set $\mathbb{E} \subseteq \mathbb{R}^m$, $m \in \mathbb{N}$. For example, if $\mathbb{E}$ is a unit box, unit ball, or unit simplex, one obtains zonotopic, ellipsoidal, or polytopic tubes. Notice that there exists a variety of set-valued integrators from the field of set-valued computing, which can be used to construct the functions C and c systematically.

These integrators can be used in combination with a suitable feedback parameterization, e.g., the affine feedback parameterization $\tilde{\mu}[K, k]$. This leads to the conservative approximation of the original robust MPC problem

$$\begin{aligned}
& \inf_{\substack{k: \mathbb{R} \rightarrow \mathbb{R}^{n_x} \\ K: \mathbb{R} \rightarrow \mathbb{R}^{n_u \times n_x}}} \int_0^T \ell(C(t, \tilde{\mu}[K, k], x_0) \cdot \mathbb{E} + c(t, \tilde{\mu}[K, k], x_0)) dt \\
& + \mathcal{M}(C(T, \tilde{\mu}[K, k], x_0) \cdot \mathbb{E} + c(T, \tilde{\mu}[K, k], x_0)) \\
& \text{s.t.} \quad \begin{cases} \forall t \in [0, T] : \\ C(t, \tilde{\mu}[K, k], x_0) \cdot \mathbb{E} + c(t, \tilde{\mu}[K, k], x_0) \subseteq \mathbb{X}, \\ K(t) (C(t, \tilde{\mu}[K, k], x_0) \cdot \mathbb{E} + c(t, \tilde{\mu}[K, k], x_0)) \subseteq \mathbb{U} \end{cases} \quad (13)
\end{aligned}$$

Depending on the particular choice of $\mathcal{M}$, ℓ , $\mathbb{X}$, and $\mathbb{U}$, this optimization problem can be discretized and processed further in order to arrive at a standard nonlinear programming problem. One remaining challenge, however, is that the resulting optimization problem is non-convex in general. Nevertheless, in principle, the above parameterization based tube MPC approach leads to practical implementations. For example in [69] a practical implementation of this approach for robust control of an exothermic reactor can be found.

Notice that there exists a great variety of variants of the above outlined approaches for Tube MPC such as Homothetic Tube MPC [66] or Elastic Tube MPC [67]. We also refer to [65, 64] for a more general overview on parameterized Tube MPC.

5.4 Tube MPC via min-max differential inequalities

One way to avoid parameterizing the feedback control law is to make use of a different parameterization of (3). Here, the main idea is to rewrite the problem in terms of so-called Robust Forward Invariant Tubes (RFITs). An RFIT is a set-valued function $\bar{X}: \mathbb{R} \rightarrow \mathbb{K}^{n_x}$ for which there exists a feedback control $\mu: \mathbb{R} \times \mathbb{R}^{n_x} \rightarrow \mathbb{U}$, such that

$$\bar{X}(t_2) \supseteq \bigcup_{x_1 \in \bar{X}(t_1)} X(t_2 - t_1, x_1, \mu)$$

for all $t_1, t_2 \in \mathbb{R}$ with $t_1 \leq t_2$. Let $\mathcal{X}$ denote the set of all RFITs for the dynamic system on $[0, T]$. The set-based MPC problem (3) can alternatively be written in the form

$$\begin{aligned}
& \inf_{\bar{X} \in \mathcal{X}} \int_0^T \ell(\bar{X}(t)) dt + \mathcal{M}(\bar{X}(T)) \\
& \text{s.t.} \quad \begin{cases} \bar{X}(t) \subseteq \mathbb{X}, \forall t \in [0, T], \\ \bar{X}(0) = \{x_0\}. \end{cases} \quad (14)
\end{aligned}$$

Now, we have traded optimizing over the feedback policy μ by an optimization problem over RFITs. Fortunately, if we restrict ourselves to input-affine nonlinear systems,

$$\dot{x}(t) = f(x(t), u(t), w(t)) = g(x(t), w(t)) + G(x(t))u(t),$$

and restrict the class of RFITs to those with compact and convex cross-sections $\bar{X}(t) \in \mathbb{K}_C^{n_x}$, we can, at least in some cases, arrive at conservative but tractable approximations of (3). The construction of such approximations requires the use of the support function

$$\forall c \in \mathbb{R}^n, \quad \sigma[Z](c) := \max_{z \in Z} c^\top z,$$

of a compact and convex set $Z \subseteq \mathbb{R}^n$. In [80] it is shown that if a set valued function $\bar{X} : \mathbb{R} \rightarrow \mathbb{K}_C^{n_x}$ satisfies for almost all $t \in [0, T]$

$$\frac{d}{dt} \sigma[\bar{X}(t)](c) \geq \min_{v \in \mathbb{U}} \max_{\xi, \omega} \left\{ c^\top f(\xi, v, \omega) \mid \begin{array}{l} c^\top \xi = \sigma[\bar{X}(t)](c) \\ \xi \in \bar{X}(t) \\ \omega \in \mathbb{W} \end{array} \right\}, \quad (15)$$

for each $c \in \mathbb{R}^{n_x}$, and the function $\sigma[\bar{X}(\cdot)](c)$ is, for all $c \in \mathbb{R}^{n_x}$, Lipschitz continuous on $[0, T]$, then it is an RFIT for the dynamic system on $[0, T]$. Thus, any solution of

$$\begin{aligned} \inf_{\bar{X}} \quad & \int_0^T \ell(\bar{X}(t)) dt + \mathcal{M}(\bar{X}(T)) \\ \text{s.t.} \quad & \begin{cases} \text{a.e. } t \in [0, T], \\ \text{Inequality (15),} \\ \bar{X}(t) \subseteq \mathbb{X}, \\ \bar{X}(0) = \{x_0\} \end{cases} \end{aligned} \quad (16)$$

is a feasible point of (14). Problem (16) is not a standard optimal control problem, as it includes a semi-infinite differential inequality constraint. However, one can use the same set parameterization strategies as in Section 5.3 in order to reformulate (16). This leads to a band-structured optimization problem, whose complexity scales linearly with the horizon length.

One of the key features of (16) is that it does not require a parameterization of the feedback law. Thus, the conservatism of the proposed approach depends only on the parameterization of the tube cross-sections $\bar{X}(t)$. In fact, in case the chosen parameterization has a smooth boundary with positive curvature, the feedback law inducing the tube is a nontrivial nonlinear function given by

$$\mu(t, y) = \mu_t^* \left(\mathcal{G}_{\bar{X}(t)}(y) \right) \quad \text{with} \quad \mu^*(c) := \underset{v \in \mathbb{U}}{\operatorname{argmin}} \ c^\top G \left(\mathcal{G}_{\bar{X}(t)}^{-1}(c) \right) v.$$

Here, $\mathcal{G}_{\bar{X}(t)} : \operatorname{bd} \bar{X}(t) \rightarrow \mathcal{S}^{n_x-1}$ denotes the Gauss map of $\bar{X}(t)$ and its inverse is given by⁹

$$\mathcal{G}_{\bar{X}(t)}^{-1}(c) = \underset{\xi \in \bar{X}(t)}{\operatorname{argmax}} \ c^\top \xi.$$

⁹ The boundary of $Z \subset \mathbb{R}^n$ is denoted by $\operatorname{bd} Z$, while $\mathcal{S}^{n-1}$ denotes the n -dimensional unit sphere.

As it stands, even the solution of a parameterized version of Problem (16) is non-trivial. But, as discussed in [79, 80], some parameterizations, e.g. ellipsoids, of the tube cross-sections, lead to practical implementations of robust MPC controllers, as well as explicit expressions for the feedback law μ .

Remark: The above min-max differential inequality uses properties of the boundary of robust forward invariant tubes of continuous-time systems, which have, at least in similar variants, been analyzed in earlier articles by Nagumo (see, e.g., [10] for a discussion of Nagumo's theorem) as well as in the context of viability theory [3]. As these boundary properties rely on differential analysis, the corresponding methods can, at least in the above form, only be applied to continuous-time systems.

6 Numerical aspects: modern set-valued computing

This section gives a concise introduction to set-valued arithmetics for factorable functions and associated tools for set-valued integration. The corresponding methods can be used as a basis for the implementation of tube-based model predictive control algorithms.

6.1 Factorable functions

A function φ is called factorable if it can be represented as a finite recursive composition of atom operations $\varphi_i \in \mathcal{L}$, with $i \in \{1, \dots, n_\varphi\}$, from a given finite library

$$\mathcal{L} = \{+, *, \sin, \exp, \log, \dots\}.$$

This library typically includes binary sums, binary products, and a number of univariate atom functions such as trigonometric functions, exponentials, as well as logarithms. In practice, factorable functions over a given library $\mathcal{L}$ are represented in the form of a computational graph, which can be obtained in most object oriented programming languages by using operator overloading or source code transformation, as visualized in Figure 3. Notice that the result a_i , which is obtained after applying an atom operation φ_i , is stored temporarily. Thus, the input arguments of the atom operations φ_i are either components of the input vector x or intermediate results of previously computed operations, i.e., components of the intermediate result vector a , as shown in the example in Figure 3. In general, the recursion can be written in the form

$$\forall k \in \{1, \dots, n_\varphi\}, \quad a \leftarrow [a, \varphi_k(x, a)]^T,$$

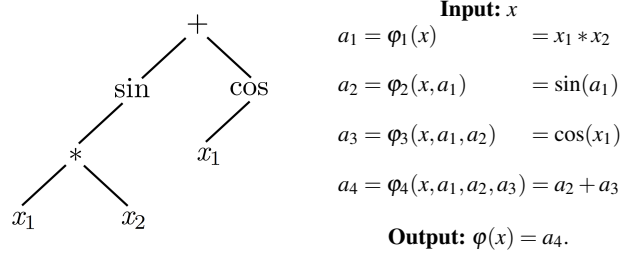


Fig. 3 Visualization of the computational graph of the function $\varphi(x) = \sin(x_1 * x_2) + \cos(x_1)$. The intermediate results a_1, a_2 , and a_3 may or may not be deleted after a function evaluation depending on whether they are needed as part of other function evaluations.

where a is initialized with the empty vector, but then the dimension of a increases by 1 after every atom evaluation.¹⁰ The function value $\varphi(x) = P_\varphi a$ with $P_\varphi \in \{0, 1\}^{n_{\text{out}} \times n_\varphi}$ corresponds to selected components of a , i.e., every row of the matrix P_φ has one entry, which is equal to 1.

Many modern computer algorithms and software are exploiting the structure of the computational graph of factorable functions explicitly, for example, algorithmic differentiation [1, 2, 30], modeling environments for convex optimization [29, 48] and optimal control [34], global optimization algorithms and software [15, 54, 71], and many set arithmetic routines [14], which are reviewed in the next section.

6.2 Set arithmetics

Let φ be a given factorable function and $\mathbb{E} \in \mathbb{K}^m$ a given basis set. The goal of a set arithmetic is to construct an enclosure function Φ of the image set map of φ with respect to the basis set $\mathbb{E}$, i.e., such that

$$\{\varphi(x) \mid x \in C \cdot \mathbb{E}\} \subseteq \Phi(C) \cdot \mathbb{E} \quad (17)$$

for any coefficient matrix $C \in \mathbb{R}^{n \times m}$. Here, we use the same notation as in Section 5.2, but we leave away the offset parameter c , since we can always redefine

$$\mathbb{E} \leftarrow \mathbb{E} \times \{1\}$$

if we want to include such offsets. Now, the main idea is to build up the function Φ recursively by passing through the computational graph of the factorable function

¹⁰ In practical implementations, the memory allocation policies for function evaluations depend on the compiler and hardware. For MPC applications on embedded hardware one often uses static memory. For example, in modern code-generation based MPC solvers for small-scale systems the memory for all components of a of all online function evaluations is pre-allocated [34].

φ . This means that we need to construct enclosure functions Φ_i for every atom operation φ_i such that

$$\forall y \in C \cdot \mathbb{E}, \quad [y^\top, \varphi_i(y)]^\top \in \Phi_i(C) \cdot \mathbb{E}.$$

for all feasible coefficient matrices $C \in \mathbb{R}^{(n_{\text{in}}+i-1) \times m}$ and all $i \in \{1, \dots, n_\varphi\}$. The enclosure function Φ is then given by the finite recursive composition

$$\Phi = P_\varphi * [\Phi_{n_\varphi} \circ \dots \circ \Phi_2 \circ \Phi_1],$$

which satisfies (17) by construction.

The key to constructing the enclosure (17) is the definition of the arithmetic rule operating on the coefficient matrix C . For example, interval arithmetics, one of the oldest and most basic set arithmetics, proceeds by defining bounding rules for binary sums and products as well as for all univariate atom operations in the given library $\mathcal{L}$. These bounding rules can for example be based on simple inclusions such as

$$\begin{aligned} [\underline{c}_1, \bar{c}_1] + [\underline{c}_2, \bar{c}_2] &\subseteq [\underline{c}_1 + \underline{c}_2, \bar{c}_1 + \bar{c}_2], \\ [\underline{c}_1, \bar{c}_1] * [\underline{c}_2, \bar{c}_2] &\subseteq [\min\{\underline{c}_1 \underline{c}_2, \underline{c}_1 \bar{c}_2, \bar{c}_1 \underline{c}_2, \bar{c}_1 \bar{c}_2\}, \max\{\underline{c}_1 \underline{c}_2, \underline{c}_1 \bar{c}_2, \bar{c}_1 \underline{c}_2, \bar{c}_1 \bar{c}_2\}], \\ e^{[\underline{c}, \bar{c}]} &\subseteq [e^{\underline{c}}, e^{\bar{c}}], \end{aligned}$$

and so on—depending on which atom operations one wants to include in $\mathcal{L}$. Finally, in order to obtain the desired parameterization for the enclosure, the endpoint representation of the interval can be converted into a midpoint-radius representation, i.e.

$$[\underline{c}, \bar{c}] = \text{mid}([\underline{c}, \bar{c}]) + \text{rad}([\underline{c}, \bar{c}])[-1, 1],$$

with

$$\text{mid}([\underline{c}, \bar{c}]) = \frac{\underline{c} + \bar{c}}{2} \quad \text{and} \quad \text{rad}([\underline{c}, \bar{c}]) = \frac{\bar{c} - \underline{c}}{2}.$$

Even though intervals are easy to store and propagate, the corresponding set arithmetics can lead to large overestimation, particularly for large values of n_φ . This wrapping effect can, however, be mitigated by using more accurate set representations. For example, rules for the construction and propagation of ellipsoids, zonotopes, polytopes, and polynomial models also exist in the literature. Table 2 presents a non-exhaustive list of different set-arithmetics together with packages implementing them.

Notice that some but not all of the set arithmetic tools in Table 2 have been developed originally for applications in control. For example, the polyhedral relaxations of the software packages BARON [71, 75] and ANTIGONE [54] have been developed in the context of solving general factorable optimization problems to global optimality, although these methods can, at least in principle, be used in the context of robust control, too. Other tools, e.g., the Ellipsoidal Toolbox [42], focus on

Set Representation	Complexity	Software	References
Intervals	$\mathbf{O}(n)$	FILIB++ [44], PROFIL [38]	[58, 59]
Ellipsoids	$\mathbf{O}(n^2)$	Ellipsoidal Toolbox [42], MC++ [14], CRONOS [15]	[40, 41] [36, 78, 81]
Zonotopes	$\mathbf{O}(nm)$	INTLAB (Affine Arithmetic) [70],	[18, 26]
Polytopes	$\mathbf{O}(nm)$	BARON [71], ANTIGONE [54] GLOMIO [53], MPT3 [32]	[75, 5]
Taylor models	$\mathbf{O}(nr^s)$	COSY INFINITY [52], MC++, CRONOS	[46, 51]
Chebyshev models	$\mathbf{O}(nr^s)$	CHEBFUN [21], MC++	[76, 62, 22]

Table 2 References about arithmetics using particular classes of basis sets (compare Table 1) and associated software tools. The second column lists the storage complexity of the used set representations, which usually coincides with the computational complexity per atom enclosure operation. However, there are some exceptions. For example, some atom operation bounding rules for ellipsoidal arithmetic may have computational complexity $\mathbf{O}(n^3)$, if dense matrix-matrix multiplications are not avoided in the implementation.

particular set operations and limited libraries $\mathcal{L}$, as ellipsoidal calculus has originally been developed in the context of reachable set computations for linear control systems [40]. For a more general overview about set theoretic methods in control (with a strong focus on linear systems) we refer to the textbook of Blanchini and Miani [10].

It is clear that none of the above affine set arithmetics yields exact enclosures, if the exact image set is not computer representable with respect to the chosen basis set. In practice, interval arithmetics often yields very conservative enclosures, and, at least, in the context of set-valued integration for nonlinear ODEs, it can be shown that ellipsoidal, zonotopic or other affinely invariant set arithmetics lead to more stable (and, consequently more accurate) enclosures as discussed in the following section and in [37]. The accuracy of more expensive set arithmetics such as Taylor models is often analyzed approximately, i.e., for sets with a “sufficiently small” diameter [11]. The analysis of the conservatism of polynomial set arithmetics on larger domains is, however, still an active field of research [22, 62].

6.3 Set-valued integrators

The solution of continuous-time ODEs is typically not factorable. This implies that the set-arithmetic methods from the previous section are not directly applicable to bound the reachable set of the closed-loop system

$$\dot{x}(t) = f(x(t), \mu(t, x(t)), w(t)) \quad \text{with} \quad x(0) = x_0,$$

which is needed in the context of tube based MPC. In order to slightly simplify the following discussion, we assume that this ODE can be written in the form

$$\dot{x}(t) = g(t, x(t), p) \quad \text{with} \quad x(0) = x_0$$

with a finite dimensional uncertain parameter $p \in \mathbb{P} \in \mathbb{K}^{n_p}$ by parameterizing the uncertain function w , e.g., by using a polynomial parameterization

$$w(t) \approx \sum_{k=0}^N p_k t^k.$$

This parameterization can be done in a rigorous manner by over-estimating the associated parameterization error and constructing the set $\mathbb{P}$ appropriately, such that

$$X(t, \mu, x_0) \subseteq \left\{ x_t \in \mathbb{R}^{n_x} \left| \begin{array}{l} \exists x \in W_{1,2}^{n_x}, \exists p \in \mathbb{P} : \forall \tau \in [0, t], \\ \dot{x}(\tau) = g(\tau, x(\tau), p) \\ x(0) = x_0, x(t) = x_t \end{array} \right. \right\},$$

as discussed in full detail in [33, 35]. Next, a local Taylor expansion of the solution trajectory of the parametric ODE can be obtained by constructing the functions

$$\begin{aligned} \chi_0(t, x, p) &= x \\ \chi_k(t, x, p) &= \frac{1}{k} \left(\frac{\partial \chi_{k-1}(t, x, p)}{\partial x} g(t, x, p) + \frac{\partial \chi_{k-1}(t, x, p)}{\partial t} \right) \end{aligned}$$

for all $k \in \{1, \dots, s+1\}$, where $s \in \mathbb{N}$ is the order of the expansion. If g is smooth and factorable, the functions χ_k are smooth and factorable, too, and can be generated automatically by using algorithmic differentiation [30]. Thus, any affine set arithmetic can be used to construct enclosure parameters $D_k \in \mathbb{R}^{n_x \times m}$ and $d_k \in \mathbb{R}^{n_x}$ of the factorable auxiliary functions χ_k

$$\{ \chi_k(t, x, p) \mid x \in C(t, \mu, x_0) \cdot \mathbb{E} + c(t, \mu, x_0), p \in \mathbb{P} \} \subseteq D_k \cdot \mathbb{E} + d_k$$

for $k \in \{0, 1, \dots, s\}$ and

$$\{ \chi_k(t, x, p) \mid t \in [0, h], x \in C(t, \mu, x_0) \cdot \mathbb{E} + c(t, \mu, x_0), p \in \mathbb{P} \} \subseteq D_{s+1} \cdot \mathbb{E} + d_{s+1}$$

for a suitable basis set $\mathbb{E} \subseteq \mathbb{K}^m$ and for a suitable step-size $h > 0$. It follows from Taylor's theorem that

$$x(t + \tau) \in C(t + \tau, \mu, x_0) \cdot \mathbb{E} + c(t + \tau, \mu, x_0)$$

with

$$C(t + \tau, \mu, x_0) = \sum_{k=0}^{s+1} D_k \tau^k \quad \text{and} \quad c(t + \tau, \mu, x_0) = \sum_{k=0}^{s+1} d_k \tau^k$$

for all $\tau \in [0, h]$ as long as $x(t) \in C(t, \mu, x_0) \cdot \mathbb{E} + c(t, \mu, x_0)$. Thus, one can construct enclosure functions C, c in a recursive way by using suitable step-sizes. This yields a continuous-time enclosure

$$X(t, \mu, x_0) \subseteq C(t, \mu, x_0) \cdot \mathbb{E} + c(t, \mu, x_0) ,$$

which is valid on the whole time horizon $t \in [0, T]$, as needed in the context of tube-based MPC having (13) in mind. Here, one remaining difficulty is how to control the step-size h during the integration. In the literature, various strategies can be found, which have been devised to deal with this problem. The first strategy proceeds by first choosing an optimistic h but then rejects the step if the over-estimation is too large. Most validated integrators, such as `COSY-INFINITY` [52], `VSPODE` [46], and `VALENCIA-IVP` [68], are based on this or variants of this strategy.¹¹ In [37] a reversed-two phase algorithm was introduced, which proceeds in a slightly different way by constructing an a parametric enclosure that is valid on the whole time horizon. The step-size is then refined in a second phase in order to reduce the step size only if the approximation error of the Taylor expansion is large compared to the overestimation that cannot be avoided anyhow due to other set arithmetic operations.

Another practical problem with set integrators is that so-called “bound explosion phenomena” may be observed. Such bound explosions occur if one uses too conservative set arithmetics such that the associated overestimation effects are growing over time. In the worst case, these wrapping effects can lead to unstable (or unreasonably conservative) bounds—even if the original ODE was perfectly stable. As it was shown in [37] stable set-integrators can be constructed for asymptotically stable systems and for set-parameterizations, if one uses set arithmetics, which are invariant under affine transformations. For example, an affine transformation of an ellipsoid or a zonotope is again an ellipsoid or zonotope. Thus, most implementations of ellipsoidal and zonotopic set-arithmetics can be expected to be invariant under affine transformation, while standard interval arithmetic does not have such a property. A generic implementation of stable set-valued integrators for a variety of set-parameterizations can be found as part of the `CRONOS` library [15].

¹¹ Early set-valued integrators, as, for example, developed by Nedialkov and Jackson [60], are not based on direct algorithmic differentiation based Taylor expansion of the solution trajectory, but more advanced Hermite-Obreschkoff integration schemes, which have the advantage that they can deal more efficiently with stiff dynamic systems. Some of the above mentioned software packages are also using more advanced integration schemes, but the basic ideas for bounding the reachable set enclosures are, nevertheless, very similar to the easy-to-implement Taylor expansion based method, which has been outlined in this section.

7 Conclusions

Although exact inf-sup feedback MPC problems are intractable in general, this chapter has reviewed a large number of practical numerical methods, which can be used to construct conservative approximations of robust MPC. For the case that the system dynamics is linear, one can either rely on methods from the field of ellipsoidal approximation and linear matrix inequalities in control or use affine disturbance parameterizations to construct highly scalable implementations based on convex optimization. Moreover, this chapter has reviewed three main classes of generic methods for robust MPC, namely, Dynamic Programming, Scenario-Tree MPC, and Tube MPC. These methods have in common that they are—at least in principle—applicable to both linear and nonlinear dynamic processes, although mature software packages for general nonlinear robust MPC are not available yet and most practical implementations still focus on specialized classes of systems, often linear process models.

The second part of this chapter had a strong focus on reviewing state-of-the-art numerical methods for Tube MPC. This focus is motivated by recent developments in the field of modern set-valued computing, or, more specifically, affine set-arithmetic and set-valued integration, which can be considered as the basis for Tube MPC based methods. While traditional tools for computing set enclosures are often based on rather conservative interval arithmetic, the trend of modern set-arithmetics goes towards using polynomial set representations, often in combination with ellipsoidal or zonotopic remainder bounds, which are invariant under affine transformations and the basis for stable reach-set integration. These new tools are opening new perspectives for Tube MPC based implementations of robust MPC, although the development of mature tools for generic robust nonlinear MPC remains an important challenge for future research.

References

1. J. Andersson, J. Åkesson, and M. Diehl. Casadi: A symbolic package for automatic differentiation and optimal control. In *Recent Advances in Algorithmic Differentiation*, pages 297–307. Springer, 2012.
2. J. Andersson, B. Houska, and M. Diehl. Towards a computer algebra system with automatic differentiation for use with object-oriented modelling languages. In *3rd International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools; Oslo; Norway; October 3*, pages 99–105. Linköping University Electronic Press, 2010.
3. J.P. Aubin. *Viability theory*. Systems & Control: Foundations & Applications. Birkhäuser Boston, Inc., Boston, MA, 1991.
4. S. Bansal, M. Chen, S. Herbert, and C.J. Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. *arXiv preprint arXiv:1709.07523*, 2017.

5. X. Bao, N. V. Sahinidis, and M. Tawarmalani. Multiterm polyhedral relaxations for non-convex, quadratically constrained quadratic programs. *Optimization Methods Software*, 24(4-5):485–504, 2009.
6. A. Bemporad, F. Borrelli, and M. Morari. Min-max control of constrained uncertain discrete-time linear systems. *IEEE Transactions on Automatic Control*, 48(9):1600–1606, 2003.
7. A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. In *Technical report, Minerva Optimization Center, Technion, Israeli Institute of Technology*, 2002.
8. D.P. Bertsekas. Dynamic programming and optimal control, 1995.
9. Jakob Björnberg and Moritz Diehl. Approximate robust dynamic programming and robustly stable mpc. *Automatica*, 42(5):777–782, 2006.
10. F. Blanchini and S. Miani. *Set-theoretic methods in control*. Systems & Control: Foundations & Applications. Birkhäuser Boston, Inc., Boston, MA, 2008.
11. A. Bompadre, A. Mitsos, and B. Chachuat. Convergence analysis of taylor models and mccormick-taylor models. *Journal of Global Optimization*, pages 1–40, 2013.
12. S Boyd, L El Ghaoui, E Feron, and V Balakrishnan. Linear matrix inequalities in system and control theory, 1994.
13. R. Buckdahn and J. Li. Stochastic differential games and viscosity solutions of hamilton-jacobi-bellman-isaacs equations. *Siam J. Control Optim.*, 47(1):444–475, 2008.
14. B. Chachuat and OMEGA Research Group. Mc++: A toolkit for construction, manipulation and bounding of factorable functions, 2006–2017. <https://omega-icl.github.io/mcpp/>.
15. B. Chachuat and OMEGA Research Group. Cronos: Complete search optimization for non-linear systems, 2012–2017. <https://omega-icl.github.io/cronos/>.
16. B. Chachuat, B. Houska, R. Paulen, N. Perić, J. Rajyaguru, and M.E. Villanueva. Set-theoretic approaches in analysis, estimation and control of nonlinear systems. *IFAC-PapersOnLine*, 48(8):981–995, 2015.
17. M. Chen, S.L. Herbert, M.S. Vashishtha, S. Bansal, and C.J. Tomlin. Decomposition of reachable sets and tubes for a class of nonlinear systems. *arXiv preprint arXiv:1611.00122*, 2016.
18. L. H. de Figueiredo and J. Stolfi. Affine arithmetic: Concepts and applications. *Numerical Algorithms*, 7(1):147–158, 2004.
19. O.H. Bosgra D.H. van Hessem. A conic reformulation of model predictive control including bounded and stochastic disturbances under state and input constraints. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 4643–4648, 2002.
20. M. Diehl and J. Bjornberg. Robust dynamic programming for min-max model predictive control of constrained uncertain systems. *IEEE Transactions on Automatic Control*, 49(12):2253–2257, 2004.
21. T. A Driscoll, N. Hale, and L. N. Trefethen. *Chebfun Guide*. Pafnuty Publications, 2014. <http://www.chebfun.org/docs/guide/>.
22. T. Dzetkulić. Rigorous integration of non-linear ordinary differential equations in Chebyshev basis. *Numerical Algorithms*, 69(1):183–205, 2015.
23. S. Engell. Online optimizing control: The link between plant economics and process control. In *10th International Symposium on Process Systems Engineering: Part A*, volume 27 of *Computer Aided Chemical Engineering*, pages 79–86. Elsevier, 2009.
24. A. F. Filippov. On certain questions in the theory of optimal control. *J. SIAM Control Ser. A*, 1(1):76–84, 1962.
25. W.H. Fleming and P.E. Souganidis. On the existence of value functions of two-player, zero-sum stochastic differential games. *Indiana Univ. Math. J.*, 38(2):293–314, 1989.
26. K. Fukuda. From the zonotope construction to the minkowski addition of convex polytopes. *Journal of Symbolic Computation*, 38(4):1261–1272, 2004.
27. M. Gerdts, R. Henrion, D. Hömberg, and C. Landry. Path planning and collision avoidance for robots. *Numerical Algebra, Control and Optimization*, 2(3):437–463, 2012.
28. P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4):523–533, 2006.
29. M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, March 2014.

30. A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia, PA, 2008.
31. L. Grüne. An adaptive grid scheme for the discrete hamilton-jacobi-bellman equation. *Numerische Mathematik*, 75(3):319–337, 1997.
32. M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. Multi-parametric toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, Zürich, Switzerland, July, 17–19 2013. <http://control.ee.ethz.ch/mpt>.
33. B. Houska and B. Chachuat. Branch-and-lift algorithm for deterministic global optimization in nonlinear optimal control. *Journal of Optimization Theory and Applications*, 162(1):208–248, 2014.
34. B. Houska, H. J. Ferreau, and M. Diehl. An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*, 47(10):2279–2285, 2011.
35. B. Houska, J.C. Li, and B. Chachuat. Towards rigorous robust optimal control via generalized high-order moment expansion. *Optimal Control Applications and Methods*, <https://doi.org/10.1002/oca.2309>, 2017.
36. B. Houska, F. Logist, J. Van Impe, and M. Diehl. Robust optimization of nonlinear dynamic systems with application to a jacketed tubular reactor. *J. Proc. Control*, 22(6):1152 – 1160, 2012.
37. B. Houska, M. E. Villanueva, and B. Chachuat. Stable set-valued integration of nonlinear dynamic systems using affine set-parameterizations. *SIAM Journal on Numerical Analysis*, 53(5):2307–2328, 2015.
38. C. Keil. Profil: Programmer’s runtime optimized fast interval library, 2009–2017. http://www.ti3.tuhh.de/keil/profil/index_e.html.
39. M.V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.
40. A. B. Kurzhanski and T. F. Filippova. On the theory of trajectory tubes—a mathematical formalism for uncertain dynamics, viability and control. In *Advances in nonlinear dynamics and control: a report from Russia*, volume 17 of *Progr. Systems Control Theory*, pages 122–188. Birkhäuser Boston, Boston, MA, 1993.
41. A. B. Kurzhanski and I. Vályi. *Ellipsoidal calculus for estimation and control*. Systems & Control: Foundations & Applications. Birkhäuser Boston, Inc., Boston, MA, 1997.
42. A. A. Kurzhanskiy and P. Varaiya. Ellipsoidal toolbox (et). In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1498–1503, 2006.
43. W. Langson, I. Chrysoschoos, S. V. Raković, and D. Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.
44. M. Lerch, G. Tischler, J. W. V. Gudenberg, W. Hofschuster, and W. Krämer. Filib++, a fast interval library supporting containment computations. *ACM Trans. Math. Softw.*, 32(2):299–324, 2006. <http://www2.math.uni-wuppertal.de/xsc/software/filib.html>.
45. D. Liberzon. *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2012.
46. Y. Lin and M. A. Stadtherr. Validated solutions of initial value problems for parametric odes. *Applied Numerical Mathematics*, 57(10):1145–1162, 2007.
47. J. Löfberg. *Minimax approaches to robust model predictive control*. PhD thesis, Linköping University, 2003.
48. J. Löfberg. Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2004.
49. Johan Löfberg. Approximations of closed-loop mpc. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 1438–1442, Maui, Hawaii, 2003.
50. S. Lucia, T. Finkler, and S. Engell. Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *Journal of Process Control*, 23(9):1306–1319, 2013.
51. K. Makino and M. Berz. Taylor models and other validated functional inclusion methods. *Int. J. Pure Appl. Math*, 4(4):379–456, 2003.

52. K. Makino and M. Berz. Cosy infinity version 9. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 558(1):346–350, 2006.
53. R. Misener and C. A. Floudas. Glomiqo: Global mixed-integer quadratic optimizer. *Journal of Global Optimization*, 57(1):3–50, 2013.
54. R. Misener and C. A. Floudas. ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. *Journal of Global Optimization*, 2014. DOI: 10.1007/s10898-014-0166-2.
55. I. M. Mitchell. The flexible, extensible and efficient toolbox of level set methods. *Journal of Scientific Computing*, 35:300–329, 2008.
56. I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.
57. I. M. Mitchell and J. A. Templeton. A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems. *Hybrid Systems Computation and Control, Lecture Notes in Computer Science*, 3414:480–494, 2005.
58. R.E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.
59. R.E. Moore, R.B. Kearfott, and M.J. Cloud. *Introduction to Interval Analysis*. SIAM, Philadelphia, PA, 2009.
60. N.S. Nedialkov and K.R. Jackson. An interval hermite-obreschkoff method for computing rigorous bounds on the solution of an initial value problem for an ordinary differential equation. In *Developments in Reliable Computing*, pages 289–310. Springer Netherlands, 1999.
61. L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. John Wiley & Sons, New York, 1962.
62. J. Rajyaguru, M. E. Villanueva, B. Houska, and B. Chachuat. Chebyshev model arithmetic for factorable functions. *Journal of Global Optimization*, 68(2):413–438, 2017.
63. S. V. Raković. Set theoretic methods in model predictive control. In *Nonlinear Model Predictive Control: Towards New Challenging Applications*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 41–54. Springer, Heidelberg, 2009.
64. S.V. Raković. Invention of prediction structures and categorization of robust MPC syntheses. *IFAC Proceedings Volumes*, 45(17):245–273, 2012.
65. S.V. Rakovic, B. Kouvaritakis, M. Cannon, C. Panos, and R. Findeisen. Parameterized tube model predictive control. *IEEE Transactions on Automatic Control*, 57(11):2746–2761, 2012.
66. S.V. Raković, B. Kouvaritakis, R. Findeisen, and M. Cannon. Homothetic tube model predictive control. *Automatica*, 48(8):1631–1638, 2012.
67. S.V. Raković, W.S. Levine, and B. Açikmeşe. Elastic tube model predictive control. In *American Control Conference (ACC), 2016*, pages 3594–3599. IEEE, 2016.
68. A. Rauh, E. P. Hofer, and E. Auer. Valencia-ivp: A comparison with other initial value problem solvers. In *12th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN 2006)*, pages 36–36, 2006.
69. J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison (WI), 5th edition, 2015.
70. S.M. Rump. INTLAB - INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. <http://www.ti3.tuhh.de/rump/>.
71. N. V. Sahinidis. BARON 14.3.1: Global Optimization of Mixed-Integer Nonlinear Programs, *User's Manual*, 2005–2017. <http://www.minlp.com/downloads/docs/baron%20manual.pdf>.
72. P.O.M. Scokaert and D.Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic control*, 43(8):1136–1142, 1998.
73. G.V. Smirnov. *Introduction to the Theory of Differential Inclusions*. Graduate Studies in Mathematics, Volume 41, American Mathematical Society, 2002.
74. E.D. Sontag. *Mathematical control theory: deterministic finite dimensional systems*. Springer Science & Business Media, 2013.
75. M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.

76. L. N. Trefethen and Z. Battles. An Extension of MATLAB to continuous functions and operators. *SIAM Journal on Scientific Computing*, 25:1743–1770, 2004.
77. B. P. G. Van Parys, P. J. Goulart, and M. Morari. Infinite horizon performance bounds for uncertain constrained systems. *IEEE Trans. Automat. Control*, 58(11):2803–2817, 2013.
78. M. E. Villanueva, B. Houska, and B. Chachuat. Unified framework for the propagation of continuous-time enclosures for parametric nonlinear ODEs. *J. Global Optim.*, 62(3):575–613, 2015.
79. M. E. Villanueva, J. C. Li, X. Feng, B. Chachuat, and B. Houska. Computing ellipsoidal robust forward invariant tubes for nonlinear mpc. In *In Proceedings of the 20th IFAC World Congress, Toulouse, France*, page 74367441, 2017.
80. M. E. Villanueva, R. Quirynen, M. Diehl, B. Chachuat, and B. Houska. Robust MPC via minmax differential inequalities. *Automatica*, 77:311–321, 2017.
81. M.E. Villanueva, J. Rajyaguru, B. Houska, and B Chachuat. Ellipsoidal arithmetic for multivariate systems. *Computer Aided Chemical Engineering*, 37:767–772, 2015.
82. Z. Wan and M.V. Kothare. An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica*, 39(5):837–846, 2003.
83. Y. Wang, B. O’Donoghue, and S. Boyd. Approximate dynamic programming via iterated bellman inequalities. *International Journal of Robust and Nonlinear Control*, 25(10):1472–1496, 2015.
84. M. N. Zeilinger, D. M. Raimondo, A. Domahidi, M. Morari, and C. N. Jones. On real-time robust model predictive control. *Automatica*, 50(3):683–694, 2014.