Chapter 13
A Set-Theoretic Method for Verifying Feasibility of a Fast Explicit Nonlinear Model Predictive Controller

Davide M. Raimondo, Stefano Riverso, Sean Summers, Colin N. Jones,
John Lygeros and Manfred Morari

Abstract In this chapter an algorithm for nonlinear explicit model predictive control is presented. A low complexity receding horizon control law is obtained by approximating the optimal control law using multiscale basis function approximation. Simultaneously, feasibility and stability of the approximate control law is ensured through the computation of a capture basin (region of attraction) for the closed-loop system. In a previous work, interval methods were used to construct the capture basin (feasible region), yet this approach suffered due to slow computation times and high grid complexity.

In this chapter, we suggest an alternative to interval analysis based on zonotopes. The suggested method significantly reduces the complexity of the combined function approximation and verification procedure through the use of DC (difference of

Davide M. Raimondo
Automatic Control Laboratory, ETH, Physikstrasse 3, 8092 Zürich, Switzerland
e-mail: davide.raimondo@control.ee.ethz.ch

Stefano Riverso
Laboratorio di Identificazione e Controllo dei Sistemi Dinamici, Università degli Studi di Pavia, Dipartimento di Informatica e Sistemistica, Via Ferrata, 1, 27100 Pavia, Italy
e-mail: stefano.riverso@unipv.it

Sean Summers
Automatic Control Laboratory, ETH, Physikstrasse 3, 8092 Zürich, Switzerland
e-mail: ssummers@control.ee.ethz.ch

Colin N. Jones
Automatic Control Laboratory, ETH, Physikstrasse 3, 8092, Zürich, Switzerland
e-mail: cjones@control.ee.ethz.ch

John Lygeros
Automatic Control Laboratory, ETH, Physikstrasse 3, 8092 Zürich, Switzerland,
e-mail: lygeros@control.ee.ethz.ch

Manfred Morari
Automatic Control Laboratory, ETH, Physikstrasse 3, 8092 Zürich, Switzerland
e-mail: morari@control.ee.ethz.ch

R. Johansson and A. Rantzer (Eds.): Distributed Decision Making and Control, LNCIS 417, pp. 289–311.
springerlink.com

convex) programming, and recursive splitting. The result is a multiscale function approximation method with improved computational efficiency for fast nonlinear explicit model predictive control with guaranteed stability and constraint satisfaction.

13.1 Introduction

This chapter proposes a method of approximate explicit model predictive control (MPC) for nonlinear systems. While it is possible to compute the optimal control law offline for a limited number of cases (e.g., affine or piecewise affine dynamics [5, 20, 29]), it is in general necessary to approximate, and therefore validation techniques are required for the resulting approximate closed-loop system. In this chapter, we present a new technique for approximation and certification of stability and recursive feasibility for explicit NMPC controllers, in which the control law is precomputed and verified offline in order to speed online computation.

The control law is approximated via an adaptive interpolation using second order interpolets, which results in an extremely fast online computation time and low data storage. The resulting suboptimal closed-loop system is verified by computing an inner approximation of the capture basin and an algorithm is proposed that iteratively improves the approximation where needed in order to maximize the size of the capture basin. The key novelty of this chapter is the use of difference of convex (DC) programming and zonotope approximation in order to significantly improve both the computational performance and efficacy of the calculation of the capture basin.

Methods for the approximation of explicit solutions of nonlinear model predictive control (NMPC) problems have been addressed recently by various authors (e.g., see [8, 19]). In [8], the authors compute an approximate control law $\tilde{u}(x)$ with a bound on the controller approximation error ($\|u^*(x) - \tilde{u}(x)\|$), from which performance and stability properties are derived using set membership (SM) function approximation theory. In [19] the authors use multiparametric nonlinear programming to compute an explicit approximate solution of the NMPC problem defined on an orthogonal structure of the state-space partition. An additional example of the approximation of explicit solutions of NMPC can be found in [26].

In almost all cases, the suboptimality of the resulting control law (and as a consequence the stability of the feedback system) is valid under various strong assumptions. Examples include the approximation of the Lipschitz constant ([8]) and the availability of global optimization tools ([19]). While these approaches often work well in practice, in many problems the stability of the closed-loop system (and the resulting region of attraction) cannot be guaranteed. Thus, in this chapter we exploit advances in reachability analysis and adaptive interpolation to construct an approximate explicit control law that encompasses the strengths of the recent works ([8, 19]) while guaranteeing stability and feasibility and preserving a minimal representation of the control law.

Extending the results of [30, 31], in this chapter we introduce a constructive algorithm for the approximation of an explicit receding horizon NMPC control law. We approximate the optimal control law by adaptive interpolation using second order interpolets, while concurrently verifying feasibility and stability of the resulting feedback system via the computation of an inner approximation of the capture basin (see, e.g., [12]). In contrast to the capture basin computational method considered in [12, 31], we develop a mechanism for computing the capture basin using zonotopes [22, 11, 33] and DC programming [3] that significantly reduces the complexity of the combined function approximation and verification procedure. Using zonotopes and DC programming rather than interval analysis [25, 6] additionally leads to an approximate control law with less storage requirements and a larger verifiable region of attraction. With the approach we propose, we are able to construct a sparse approximation of the optimal control law while taking into consideration performance loss and the feasibility and stability of the feedback system. Further, since the solution is defined on a gridded hierarchy, the online evaluation of the control law is extremely fast, see [30].

The rest of the chapter is arranged as follows: Section 13.2 introduces the NMPC problem. Section 13.3 provides background on multiscale sparse function approximation and Secs. 13.4 and 13.5 discuss reachability analysis and the proposed method of calculating the capture basin of an approximation NMPC explicit control law. Section 13.6 provides a numerical example indicating the effectiveness of the approach.

13.2 Nonlinear Model Predictive Control

Consider the following finite horizon optimal control problem (NMPC):

$$\begin{aligned} J^*(x) = \min_{u_0,\dots,u_{N-1}} \quad & J(u_0,\dots,u_{N-1},x_0,\dots,x_N) \\ \text{subject to} \quad & x_{i+1} = f(x_i,u_i), \quad \forall i = 0,\dots,N-1 \\ & (x_i,u_i) \in \mathcal{X} \times \mathcal{U}, \quad \forall i = 0,\dots,N-1 \\ & x_N \in \mathcal{X}_F, \\ & x_0 = x, \end{aligned} \tag{13.1}$$

where $x_i \in \mathbb{R}^n$ is the state of the system, $u_i \in \mathbb{R}^m$ is the control input of the system, and N is the prediction horizon length. The cost function J takes the form

$$J(u_0,\dots,u_{N-1},x_0,\dots,x_N) := V_N(x_N) + \sum_{i=0}^{N-1} L(x_i,u_i) \ , \tag{13.2}$$

where L is the running (stage) cost and V_N is the terminal cost.

The system dynamics $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a continuous and differentiable function, and the objective is to regulate the state of the system to the origin under state and control input constraints represented by the (compact) sets $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{U} \subseteq \mathbb{R}^m$, respectively. We assume that the terminal set $\mathcal{X}_F$ is compact and positively control invariant under a known stabilizing feedback law κ_F. For the sake of simplicity (as in [8, 19]), it is assumed that the control input constraint set $\mathcal{U}$ is convex, although the following results can be extended to the nonconvex control input constraint setting.

A dual-mode NMPC control approach is taken, in which the optimal NMPC control law $\kappa^\star(x)$ is defined as

$$\kappa^\star(x) := \begin{cases} \kappa_F(x), & \text{if } x \in \mathbb{X}_F \\ u_0^\star(x), & \text{otherwise} \end{cases} \tag{13.3}$$

where $u^*(x) = u_0^\star(x), \ldots, u_{N-1}^\star(x)$ is an optimal sequence of inputs of NMPC problem (13.1) for the state x. Following [8, 19], we assume in this chapter that the optimal receding horizon control law $u_0^*(x)$ asymptotically stabilizes the origin of the closed-loop system.

Remark 13.1. Note that the proposed approximation and analysis techniques can be applied to any optimal control problem that generates a smooth control law. We here use the MPC cost function given in (13.2) because it is a common target for approximation and because sufficiently fine approximation will result in a stabilizing control law by construction.

13.3 Multiscale Function Approximation

The method we propose for approximating $u_0^\star(x)$ relies on coarsely gridding the state space, and then regridding with increasing resolution only the regions which have not been approximated sufficiently. At the same time, we keep only the grid points that play a significant role in the function approximation [30].

Define the one-dimensional scaling function with support $[-1,1]$ by

$$\phi(x) := \begin{cases} 1 - |x| \ , & \text{if } x \in [-1,1], \\ 0 \ , & \text{otherwise.} \end{cases} \tag{13.4}$$

In one dimension, we consider a dyadic discretization on the unit interval $\Omega = [0,1]$. The resulting grid Ω_l is characterized by the level of discretization l and the index i. At level l the distance between points is $h_l = 2^{-l}$ and the number of points is $N = 2^l + 1$. The index i determines the location of the grid points according to the equation

$$x_{l,i} := i \cdot h_l, \ \ 0 \leq i \leq 2^l.$$

Given function (13.4), via translation and dilation we get

$$\phi_{l,i}(x) = \phi\left(\frac{x - i \cdot h_l}{h_l}\right) \tag{13.5}$$

where $\phi_{l,i}$ represents a family of basis functions with support $[x_{l,i} - h_l, x_{l,i} + h_l]$. The family of univariate multiscale functions $\psi_{l,i}$ that make up the hierarchical basis is given as

$$\psi_{l,i} = \phi_{l,i}, i \in I_l$$

where

$$I_l = \begin{cases} \{i \in \mathbb{N}_0 | 1 \le i \le 2^l - 1, i \text{ odd}\}, & l > l_0, \\ \{i \in \mathbb{N}_0 | 0 \le i \le 2^l\}, & l = l_0. \end{cases}$$

A multivariate multiscale basis on the unit cube $\Omega^d = [0,1]^d$, where d is the dimension, can be constructed by tensor product expansion of the one-dimensional multivariate functions $\psi_{l,i}$, i.e.

$$\psi_{l,\mathbf{i}} = \prod_{j=1}^{d} \psi_{l,i_j} \tag{13.6}$$

with the d-dimensional multi-index $\mathbf{i} \in I_l^d$ and

$$I_l^d = \begin{cases} \{\mathbf{i} \in \mathbb{N}_0^d | \mathbf{0} \le \mathbf{i} \le \mathbf{2}^l\} \setminus \{\mathbf{i} \in \mathbb{N}_0^d | \mathbf{0} \le \mathbf{i} \le \mathbf{2}^l, i_j \text{ even } \forall j \in [1,d]\} & l > l_0 \\ \{\mathbf{i} \in \mathbb{N}_0^d | \mathbf{0} \le \mathbf{i} \le \mathbf{2}^l\} & l = l_0 \end{cases}$$

I_l^d is the full grid less those points seen at previous levels, as depicted in Fig. 13.1. The d-dimensional hierarchical function spaces of piecewise d-linear functions can be defined as $W_l^d = \text{span}\{\psi_{l,\mathbf{i}} : \mathbf{i} \in I_l^d\}$. Let

$$\phi_{l,\mathbf{i}}(x) = \prod_{j=1}^{d} \phi_{l,i_j}(x_j)$$

the family of d-dimensional nodal basis functions and $V_l^d = \text{span}\{\phi_{l,\mathbf{i}} : \mathbf{0} \le \mathbf{i} \le \mathbf{2}^l\}$ the d-dimensional nodal function space. It holds that $V_l^d = \bigoplus_{k \le l} W_k^d$ where $\bigoplus$ denotes the direct sum. Therefore, any function $u_l \in V_l^d$ can be represented in the hierarchical basis by

$$u_l(x) = \sum_{k=l_0}^{l} \sum_{\mathbf{i} \in I_k^d} w_{k,\mathbf{i}} \cdot \psi_{\mathbf{k},\mathbf{i}}(x)$$

where coefficients $w_{k,\mathbf{i}} \in \mathbb{R}$ (hierarchical details) correspond to the difference between the true function value $u_l(x_{k,\mathbf{i}})$ and the value of the approximate function one level below.

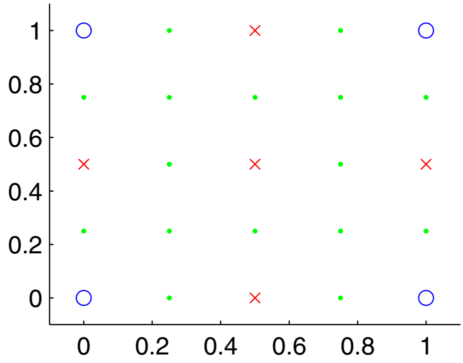

Fig. 13.1 Grid points for subspaces W_0^2 (circles), W_1^2 (x's), and W_2^2 (dots).

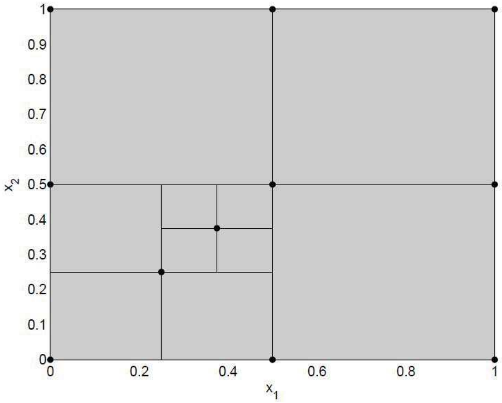

Fig. 13.2 Set of hypercubic regions $\bar{R}$.

In [30], it has been shown that the function approximation by adaptive hierarchical basis function expansion generates a grid (of hypercubes (Fig. 13.2)) spanned by an interpolation by barycentric coordinates. Given a set $\bar{R}$ of hypercubic regions, for each hypercube R it holds that

$$\hat{u}(x) = \sum_{v \in \text{extr}(R)} \hat{u}(v) f_v(x), \qquad \text{if } x \in R. \tag{13.7}$$

where extr(R) are the extreme points of R and $f_v(x)$ are compactly supported basis functions of the form (13.6) centered at the corners of the hypercube R.

Theorem 13.1. *Given any hypercube R of $\bar{R}$, if $\hat{u}(v) \in U$, $\forall v \in extr(R)$ and U is convex, then $\hat{u}(x) \in U$, $\forall x \in R$. Moreover, if problem (13.1) is convex, then, feasibility* *at vertexes v of R is necessary and sufficient in order to get feasibility for all $x \in R$, i.e., $g(x,\hat{u}(x)) \leq 0$, $\forall x \in R$.*

Proof: The result is obtained by exploiting the barycentricity of the interpolation. See [30] for details.

Note that, in the general nonlinear case, the constraint satisfaction at the vertexes of R is not sufficient in order to prove their satisfaction in the entire box. This is because g and h are in general nonconvex. However, if U is convex, then control constraint satisfaction at the vertexes of R guarantees their satisfaction in R.

Summarizing, we need an alternative method for verifying the stability and state constraints satisfaction of system $x_{i+1} = f(x_i,u_i)$ in closed loop with $\hat{u}(x)$. In the following section, we present reachability analysis of nonlinear systems as a possible solution.

13.4 Reachability

The exact computation of the set

$$\Phi = f(\Omega) \tag{13.8}$$

given an initial set $\Omega \subseteq \mathbb{R}^n$ and a map $f : \mathbb{R}^n \to \mathbb{R}^n$, is not possible in general. Taking this into account, the objective is to construct an outer approximation of Φ in such a way that the set is representable on a computer and the overestimation is kept as small as possible [22]. Several solutions have been proposed in the literature, and in the following sections we provide an overview of existing methods that apply for nonlinear functions f before proposing a new method for guaranteeing tight overbounds.

13.4.1 Interval Arithmetic

Given $S_1, S_2 \subseteq \mathbb{R}^n$ the Minkowski sum is defined as $S_1 \oplus S_2 = \{x+y : x \in S_1, y \in S_2\}$. Given $a,b \in \mathbb{R}$, with $a \leq b$, $[a,b]$ denotes the interval $\{x : a \leq x \leq b\}$. The center of the interval $[a,b]$ is denoted by $\text{mid}([a,b]) = \frac{a+b}{2}$. Let $\mathbb{I}$ be the set of all intervals $[a,b]$, i.e. $\mathbb{I} = \{[a,b] : a,b \in \mathbb{R}, a \leq b\}$. The set of all interval vectors in $\mathbb{R}^n$ is denoted by $\mathbb{I}^n$. The unitary interval $[-1,1]$ is denoted by $\mathbf{B}$. A box is an interval vector and a unitary box, denoted by $\mathbf{B}^m$, is a box composed of m unitary intervals. With a slight abuse of notation, when the superscript is not indicated, $\mathbf{B}$ denotes a generic unitary box. Given a box $\mathbf{X} = [a_1,b_1] \times [a_2,b_2] \ldots \times [a_n,b_n]$, $\text{mid}(\mathbf{X}) = (\text{mid}([a_1,b_1]),\ldots,\text{mid}[a_n,b_n])^T$ denotes the midpoint (or center) of $\mathbf{X}$, $\text{diam}(\mathbf{X}) = (b_1 - a_1,\ldots,b_n - a_n)^T$ and $\text{rad}(\mathbf{X}) = \text{diam}(\mathbf{X})/2$. Interval arithmetic is based on operations applied to intervals. An operation $\bullet$ can be extended from real numbers to intervals, i.e., given $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{I}$, $\mathbf{X}_1 \bullet \mathbf{X}_2 = \{x_1 \bullet x_2 : x_1 \in \mathbf{X}_1, x_2 \in \mathbf{X}_2\}$. The four basic interval operations as well as the interval extension of standard functions $(\sin, \cos, \tan, \arctan, \exp, \ln, |,| \text{sqrt})$, are defined in [25].

Definition 13.1. (Natural interval extension [21]) If $f : \mathbb{R}^n \to \mathbb{R}^n$ is a function computable as an expression, algorithm or computer program involving the four elementary arithmetic operations and standard functions, then a *natural interval extension* of f, denoted by $\Box f$, is obtained by replacing the occurrence of each variable by the corresponding interval variable and by executing all operations.

Theorem 13.2 ([21]). *Given a function $f : \mathbb{R}^n \to \mathbb{R}^n$ and any box $\mathbf{X} \subseteq \mathbb{R}^n$ within the domain of f, a natural interval extension $\Box f : \mathbb{I}^n \to \mathbb{I}^n$ of f satisfies $f(\mathbf{X}) \subseteq \Box f(\mathbf{X})$.*

Definition 13.2. (Taylor interval extension of degree k) Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a $k+1$ times differentiable function, $\mathbf{X} \subseteq \mathbb{R}^n$ any box within the domain of f and $y \in \mathbf{X}$. The *Taylor interval extension of f of degree k* is given by

$$\Box^k f(\mathbf{X}) = \sum_{i=0}^{k} \frac{1}{i!} \nabla^i f(y) \cdot (\mathbf{X} - y)^i + \Box r_k(\mathbf{X}, \mathbf{X}, y)$$

where $\nabla^i f(y)$ is the ith order differential of f at the point y and $\Box r_k$ is an interval extension of the Taylor remainder

$$r_k(x, \xi, y) = \frac{1}{(k+1)!} \nabla^{k+1} f(\xi) \cdot (x - y)^{k+1}.$$

By substituting $\mathbf{X}$ for ξ we obtain an overestimation of the remainder. Usually, y is chosen to be the midpoint of the box $\mathbf{X}$, and natural interval extension is used to bound the remainder.

Theorem 13.3. *Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a $k+1$ times differentiable function and $\mathbf{X} \subseteq \mathbb{R}^n$ any box within the domain of f. A Taylor interval extension of f of degree k satisfies $f(\mathbf{X}) \subseteq \Box^k f(\mathbf{X})$.*

Because of the special form of r_k, in practice the Taylor remainder usually decreases as $|x-y|^{k+1}$. Hence if $|x-y|$ is chosen to be small, then the interval extension of the Taylor remainder gets smaller for increasing k, i.e., higher order Taylor interval extensions yield better enclosures on small boxes [24]. A comparison between natural interval extension and Taylor interval extension of degree 0 (yellow) and 9 (green) is given in Figs. 13.3 and 13.4. While Fig. 13.3 shows the advantage of Taylor interval extension on a small box, Fig. 13.4 shows its limits over big boxes. In this work, interval arithmetic has been implemented using INTLAB [27].

The main drawback of interval analysis is that it always outer bounds the image of a box with a box. The use of more complex domains can reduce the conservatism. For this reason, the use of zonotopes as the class of approximates is considered in the next section.

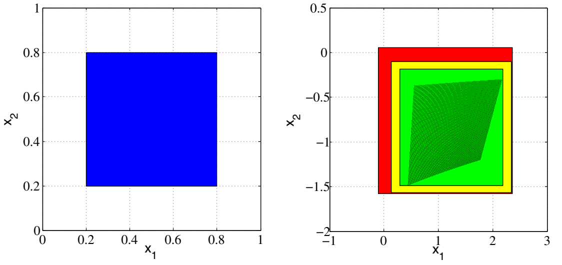

Fig. 13.3 In this example we considered the function f defined in (13.14). The starting set $\mathbf{X}$ is depicted on the left. On the right, samples of $f(\mathbf{X})$ are depicted in black while the red, yellow and green boxes represent the natural interval extension, the Taylor interval extension of degree 0 and 9 ($y = \text{mid}(\mathbf{X})$) respectively.

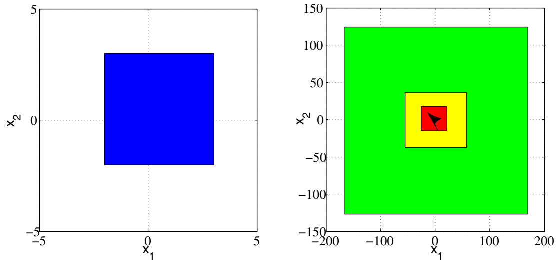

Fig. 13.4 In this example we considered the function f defined in (13.14). The starting set $\mathbf{X}$ is depicted on the left. On the right, samples of $f(\mathbf{X})$ are depicted in black while the red, yellow and green boxes represent the natural interval extension, the Taylor interval extension of degree 0 and 9 ($y = \text{mid}(\mathbf{X})$) respectively.

13.4.2 Zonotopes

Zonotopes are centrally symmetric convex polytopes. Given a vector $p \in \mathbb{R}^n$ and a matrix $H \in \mathbb{R}^{n\times m}$, the zonotope $\mathbf{Z}$ of order $n \times m$ is the set

$$\mathbf{Z} = p \oplus H\mathbf{B}^m = \{p + Hz | z \in \mathbf{B}^m\}.$$

The zonotope $\mathbf{Z}$ is the Minkowski sum of the line segments defined by the columns of the matrix H translated to the central point p. $\mathbf{Z}$ can be described as the set spanned by the column vectors of H

$$\mathbf{Z} = \{p + \sum_{i=1}^{m} \alpha_i h_i | -1 \leq \alpha_i \leq 1\}$$

where h_i, also called the *line segment generator*, is the ith column of H. When the matrix H is diagonal, the zonotope is a box composed of n intervals. The construction of zonotopes based on the Minkowski addition of convex polytopes is described in [16] and here adopted and implemented using the MPT toolbox [23]. An example of a two-dimensional (2D) zonotope is depicted in Fig. 13.5.

The image of a zonotope $\mathbf{Z}$ under a nonlinear function is not, in general, a zonotope. Kühn developed a procedure that guarantees a tight approximation by bounding $f(\mathbf{Z})$ with a zonotope [22]. The following theorem introduces the zonotope inclusion operator that is needed for computing Kühn's zonotope extension.

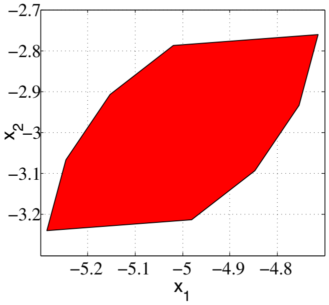

Fig. 13.5 A 2D zonotope.

Theorem 13.4. *(Zonotope inclusion) Consider a family of zonotopes represented by* $\mathbf{Z} = p \oplus \mathbf{M}\mathbf{B}^m$*, where* $p \in \mathbb{R}^n$ *is a real vector and* $\mathbf{M} \in \mathbb{I}^{n\times m}$ *is an interval matrix. A zonotope inclusion, denoted by* $\diamond(\mathbf{Z})$*, is defined by*

$$\diamond(\mathbf{Z}) = p \oplus [mid(\mathbf{M}) \quad G] \begin{bmatrix} \mathbf{B}^m \\ \mathbf{B}^n \end{bmatrix} = p \oplus J\mathbf{B}^{m+n},$$

where $G \in \mathbb{R}^{n\times n}$ *is a diagonal matrix that satisfies*

$$G_{ii} = \sum_{j=1}^{m} \frac{diam(\mathbf{M}_{ij})}{2}, \; i = 1,\ldots,n.$$

Under these definitions it results that $\mathbf{Z} \subseteq \diamond(\mathbf{Z})$.

Theorem 13.5. *(Zonotope extension) Consider a function* $f : \mathbb{R}^n \to \mathbb{R}^n$ *with continuous derivatives and a zonotope* $\mathbf{Z} = p \oplus H\mathbf{B}^m$*. Given an interval matrix* $\mathbf{M} \in \mathbb{I}^{n\times m}$ *such that* $\mathbf{M} \supseteq \nabla f(\mathbf{Z})H$*, it results that*

$$f(\mathbf{Z}) \subseteq f(p) \oplus \diamond(\mathbf{M}\mathbf{B}^m).$$

This theorem is a particular case of Kühn's method (see Proof in [2]). Note that, $\nabla f(\mathbf{Z})H$, multiplication of a matrix of sets by a matrix, is a matrix of sets. A possible outer bound is $\mathbf{M} = \Box\nabla f(\Box\mathbf{Z})H$.

The zonotope extension represents a Taylor extension of order 0, where the remainder has been evaluated on $\mathbf{Z}$. A comparison between zonotope Taylor extension of order 0 and Taylor interval extension of order 0 is given in Fig. 13.6. As expected, zonotopes better approximate the output set.

A first order Taylor zonotope extension was proposed in [11]

$$f(\mathbf{Z}) \subseteq f(p) \oplus \nabla f(p)(\mathbf{Z} - p) \oplus c_R \oplus [Z_Q \; Z_H]\mathbf{B}$$

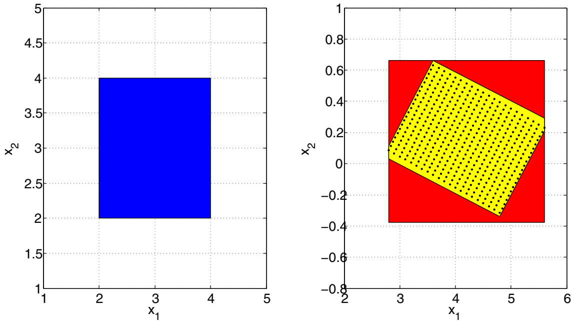

Fig. 13.6 In this example we considered the function f defined in (13.14). The starting set $\mathbf{Z}$ is depicted on the left. On the right, samples of $f(\mathbf{Z})$ are depicted in black while the red and yellow zonotopes represent the Taylor interval extension of degree 0 and the zonotope extension of degree 0 respectively.

where c_R and $[Z_Q \; Z_H]$ provide a less conservative approximation of the remainder by making use of some interval analysis properties (see Eq. (14) in [11] for the definition of c_R and $[Z_Q \; Z_H]$).

A Taylor zonotope extension of order k can been obtained as follows:

$$f(\mathbf{Z}) \subseteq f(p) \oplus \nabla f(p)(\mathbf{Z}-p) \oplus c_R \oplus Z_Q\mathbf{B} \oplus \sum_{i=3}^{k} \frac{1}{i!} \nabla^i f(p) \cdot (\Box\mathbf{Z}-p)^i \oplus \Box r_k(\Box\mathbf{Z}, \Box\mathbf{Z}, p).$$

Note that $c_R \oplus Z_Q\mathbf{B}$ represents the second order Taylor expansion term computed at p, center of the zonotope $\mathbf{Z}$ (see [11] for details). The higher order terms have been obtained by applying the Taylor interval extension. Due to over-approximation of zonotopes with boxes (i.e. $(\mathbf{Z}-p)^i$ is replaced by $(\Box\mathbf{Z}-p)^i$), there is no guarantee that higher order Taylor zonotope extensions will produce tighter enclosures than low order ones.

A comparison between zonotope Taylor extension of order 0 (red) and order 1 (yellow) (computed as in [11]) is given in Figs. 13.7 and 13.8. In Fig. 13.7, as one would expect, the first-order extension is better, but in Fig. 13.8 it is the opposite. This depends on the dynamics f (Eq. (13.13) in the case of Fig. 13.7 and Eq. (13.14) in Fig. 13.8) and the size of the starting set $\mathbf{Z}$, i.e., the larger the set, the worse the approximation. Furthermore, the use of higher order extensions does not guarantee improvement a priori. In Figs. 13.9 and 13.10, starting sets that are boxes (and not generic zonotopes) have been considered. In the first figure, higher-order extensions do better while in the second the opposite is observed. Again, this depends on the dynamics f and the size of the starting set $\mathbf{Z}$.

It is important to note that if a linear system is considered, zonotopes provide an exact description of the reachable set, unless a bound on the number of line generators is imposed. If the system is instead nonlinear, and the starting set is a small box, then high order extensions are generally better than low order ones, although the dynamics of the system plays an important role in determining which approach provides the best approximation. The drawback of high-order extensions is the need to compute the derivatives $\nabla^i f, \; i = 1, \cdots, k+1$. For this reason, we are interested in finding other techniques that are less computationally expensive while still rea-

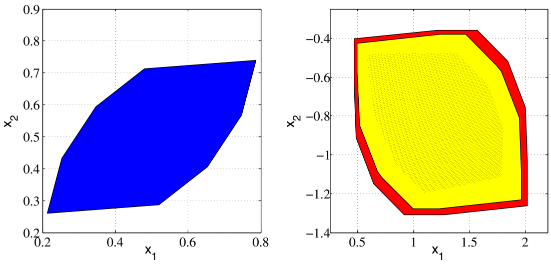

Fig. 13.7 In this example we considered the function f defined in (13.13). The starting zonotope $\mathbf{Z}$ is depicted on the left. On the right, samples of $f(\mathbf{Z})$ are depicted in black while the red and yellow zonotopes represent the Taylor zonotope extension of degree 0 and 1 (computed as in [11]) respectively.

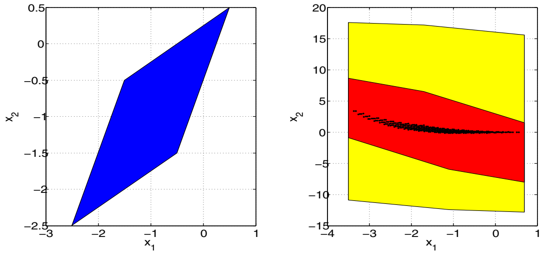

Fig. 13.8 In this example we considered the function f defined in (13.14). The starting zonotope $\mathbf{Z}$ is depicted on the left. On the right, samples of $f(\mathbf{Z})$ are depicted in black while the red and yellow zonotopes represent the Taylor zonotope extension of degree 0 and 1 (computed as in [11]), respectively.

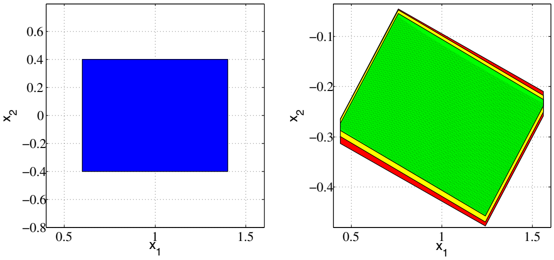

Fig. 13.9 In this example we considered the function f defined in (13.14). The starting box $\mathbf{Z}$ is depicted on the left. On the right, samples of $f(\mathbf{Z})$ are depicted in black while the red, yellow and green zonotopes represent the Taylor zonotope extension of degree 0, 1 (computed as in [11]), and 5, respectively.

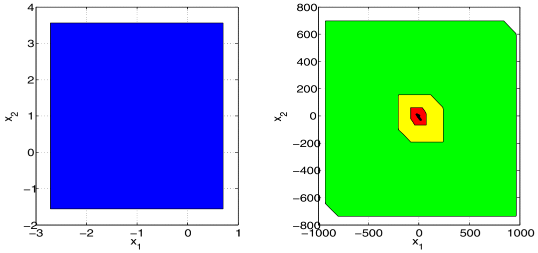

Fig. 13.10 In this example we considered the function f defined in (13.13). The starting box $\mathbf{Z}$ is depicted on the left. On the right, samples of $f(\mathbf{Z})$ are depicted in black while the red, yellow and green zonotopes represent the Taylor zonotope extension of degree 0, 1 (computed as in [11]), and 8, respectively.

sonably good in approximating the real set. In next section, we consider a method based on DC programming that provides a better first order zonotope Taylor extension than the one proposed in [11] when the function f is C^2.

13.4.2.1 DC Programming

Definition 13.3 (DC function). [17] Let Ω be a convex subset of $\mathbb{R}^n$. A real-valued function $f : \Omega \to \mathbb{R}$ is called *DC (difference of convex)* on Ω, if there exist two convex functions $g, h : \Omega \to \mathbb{R}$ such that f can be expressed in the form

$$f(x) = g(x) - h(x).$$

If $\Omega = \mathbb{R}^n$, then f is simply called a *DC function*.

Every continuous function can be approximated by a difference of two convex functions (DC functions) ([17]) and every C^2-function is a DC function ([32]). Even if f is a C^2-function, finding a DC decomposition of f—namely, the functions g and h—is often a challenging problem, although suitable procedures exist. In this work we use the method described in [1]. Given $f : \Omega \to \mathbb{R}$ and recalling that a C^2-function is convex in Ω if and only if $\nabla^2 f(x) \geq 0, \forall x \in \Omega$, we search for a parameter $\alpha \geq 0$ such that $\nabla^2 f(x) > -2\alpha I, \forall x \in \Omega$. Then $f(x) = g(x) - h(x)$ is a DC function with $g(x) = f(x) + \alpha x^T x$ and $h(x) = \alpha x^T x$.

Programming problems dealing with DC functions are called *DC programming problems*. In [3] the authors propose a DC programming-based method for constructing a tight outer approximation of $f(\mathbf{Z})$ (where f is a nonlinear C^2 function and $\mathbf{Z}$ a zonotope). First they linearize f and compute the image of $\mathbf{Z}$ under this linearization. A parallelotope that bounds the approximation error between f and its linearization for the given set $\mathbf{Z}$ is then added to the image. By exploiting the convexity of $g(x)$ and $h(x)$, a tighter approximation than the one achievable by simply bounding the remainder of the first-order Taylor approximation is thus obtained. The results are summarized in the following.

Definition 13.4 ([3]). Let $f(x) : \mathbb{R}^n \to \mathbb{R}^n$ be a nonlinear differentiable function and $\mathbf{Z} = p \oplus H\mathbf{B}^m$ a zonotope. Given $f^L(x) = f(p) + \nabla f(p)(x - p)$, the error set ε is defined as

$$\varepsilon = \{e \in \mathbb{R}^n : e = f(x) - f^L(x), x \in \mathbf{Z}\}.$$

Lemma 13.1 ([3]). *Let $\mathbf{Z} = p \oplus H\mathbf{B}^m$ be a zonotope and $f(x) : \mathbb{R}^n \to \mathbb{R}^n$ a nonlinear DC function, i.e. $f(x) = g(x) - h(x)$, with g and h convex. Let $g_i^L(x) = g(p) + \nabla g(p)(x - p)$, $h_i^L(x) = h(p) + \nabla h(p)(x - p)$ and define the parallelotope $\bar{\varepsilon}$ as*

$$\bar{\varepsilon} = \{x \in \mathbb{R}^n : \gamma_i^- \leq x_i \leq \gamma_i^+, \quad i = 1, \ldots, n\}, \tag{13.9}$$

with

$$\gamma_i^+ = \max_{x \in \mathbf{Z}} (g_i(x) - h_i^L(x) - f_i^L(x))$$
$$\gamma_i^- = \min_{x \in \mathbf{Z}} (g_i^L(x) - h_i(x) - f_i^L(x)).$$

Then, the parallelotope $\bar{\varepsilon}$ is an outer bound of the set ε, i.e., $\varepsilon \subseteq \bar{\varepsilon}$.

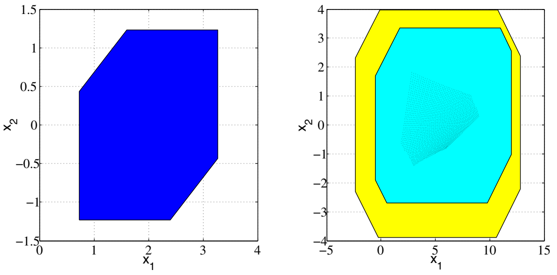

Fig. 13.11 Comparison of a DC zonotope (cyan) with a 1^{st} order one (yellow, computed as in [11]) calculated for a big starting box (blue). The black dots represent sampling of $f(\mathbf{Z})$, with f defined in (13.14).

Theorem 13.6 ([3]). *Let $\mathbf{Z} = p \oplus H\mathbf{B}^m$ be a zonotope and $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ a DC function. Then*

$$f(\mathbf{Z}) \subseteq f^L(\mathbf{Z}) \oplus \bar{\varepsilon}.$$

Theorem 13.6 is the main result of [3] and shows how to bound the image of a zonotope under a nonlinear DC function. A comparison betweeen DC programming (cyan) and first-order Taylor zonotope extension (yellow, computed as in [11]) is provided in Fig. 13.11. In this case, the DC programming-based approach produces a better approximate. Several experiments confirmed this tendency.

Summarizing, DC programming is generally better than first order Taylor zonotope extension. If the starting set is small, then high order zonotope extensions could be better. The main drawbacks of the latter approach are the computational complexity and the efficacy just for small starting sets. The reason is that Taylor approximation is a local property, i.e., it is accurate around mid$(\mathbf{Z})$ but gets worse as soon as the set becomes large. In any case, zonotopes have an intrinsic limitation in that, being symmetric and convex, they are poor at bounding nonconvex sets, while, in general, the output of $f(\mathbf{Z})$ is nonconvex.

In this chapter, we use reachability analysis in order to evaluate the image of the sets $R \in \bar{R}$ introduced in Sec. 13.3 through the nonlinear dynamics of the system in closed-loop with the approximate MPC control law. What is important to note is that R can be arbitrarily big a priori. For this reason, in the next section we propose a method for partitioning R in subsets. A similar idea has been proposed in [13]. We apply DC programming to each one of the subsets and the union of the output sets is used to contain $f(R)$. This approach should provide better performance because the starting sets are smaller and the output, being a union of zonotopes (hence not necessarily convex anymore), could be a tighter approximation to the real set. DC programming is preferred to high order zonotope extensions for computational reasons.

13.4.3 Splitting the Starting Set

First of all, we introduce the operator bisect$_k(\cdot)$ (see [33] for details) that splits a zonotope into two. Given $\mathbf{Z} = p \oplus H\mathbf{B}^m$ the operator bisect$_k(\mathbf{Z})$ generates two

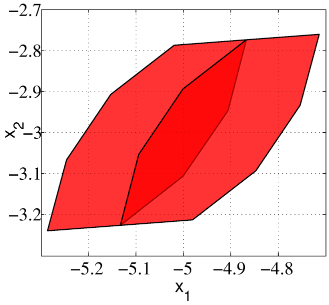

Fig. 13.12 An example of a split. bisect($\mathbf{Z}$).

subzonotopes

$$\mathbf{Z}^L = (p - \frac{h_k}{2}) \oplus [h_1 \dots \frac{h_k}{2} \dots h_m]\mathbf{B}^m$$

$$\mathbf{Z}^R = (p + \frac{h_k}{2}) \oplus [h_1 \dots \frac{h_k}{2} \dots h_m]\mathbf{B}^m$$

where h_k is the kth column of H. Figure 13.12 shows an example of the operator bisect$_k(\mathbf{Z})$ applied to the zonotope $\mathbf{Z}$ in Fig. 13.5. In this case, $\mathbf{Z}^L$ and $\mathbf{Z}^R$ intersect. This is because the line segment generators $h_1, \dots, h_m$ are not linearly independent. If $H \in \mathbb{R}^{m \times m}$, with rank$(H) = m$, then, bisect$_k(\mathbf{Z})$ provides two subzonotopes that do not overlap. As stated above, given a generic nonlinear function f, zonotope extensions as well as DC programming work better on smaller boxes. This is because Taylor approximation is a local property, i.e., it is effective for a neighborhood of mid$(\mathbf{Z})$. How big this neighborhood is depends on how close to linear the system is. For this reason, the approach we propose consists of splitting more where the system in more nonlinear, i.e., where the Taylor approximation is less effective. We evaluate this by considering $\bar{\varepsilon}$, the remainder of the first order extension with the DC programming-based approach. The procedure is summarized in Algorithm 13.1. A comparison between DC programming with $\mathbf{Z}$ split into 5 and 10 subzonotopes is reported in Table 13.1 and is depicted in Fig. 13.13. The case with 10 zonotopes produces, as expected, a tighter approximation (smaller volume) at the expense of a higher computational time and a higher number of evaluations.

Table 13.1 Algorithm 13.1. Comparison between the case of 5 and 10 splits. The algorithm has been applied in combination with the DC programming-based approach.

	# of $\mathbf{Z}_i$ in output	# of evaluated $\mathbf{Z}_i$	Comp. time (s)	Initial Volume	Final Volume
DC prog.	5	17	2.28	181.1	6.47
DC prog.	10	37	5.01	181.1	3.06

Algorithm 13.1 Splitting the starting set by using the $\bar{\varepsilon}$ criteria

```
Require: Nonlinear function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, a starting set $\mathbf{Z}$ and $n_s$, number of required splits.
Ensure: Union of $n_s$ zonotopes, which is an outer approximation of $f(\mathbf{Z})$.
Compute $\mathbf{Z_O}$, outer approximation of $f(\mathbf{Z})$ using DC programming.
$Stack = \mathbf{Z}$, $Stack_{out} = \mathbf{Z}_O$.
while $length(Stack_{out}) \leq n_s$ do
    Find in $Stack_{out}$ the zonotope $\mathbf{Z}_{O_i}$ with the biggest $\bar{\varepsilon}$ (see (13.9) for the definition) in terms of volume.
    Select from $Stack$ $\mathbf{Z}_i = p_i \oplus H_i \mathbf{B}^p$, the starting zonotope associated with $\mathbf{Z}_{O_i}$.
    for $j = 1$ to $p$ do
        Generate $\mathbf{Z}^L_{ij}$ and $\mathbf{Z}^R_{ij}$ by applying the operator bisect$(\cdot)$ to the $j$th column of matrix $H_i$.
        Compute $\mathbf{Z}^L_{O_{ij}}$ and $\mathbf{Z}^R_{O_{ij}}$, outer approximations of $f(\mathbf{Z}^L_{ij})$ and $f(\mathbf{Z}^R_{ij})$.
        Calculate the volumes of $\bar{\varepsilon}^L_{ij}$ and $\bar{\varepsilon}^R_{ij}$.
    end for
    Find the index $j$ with the smallest sum of volumes of $\bar{\varepsilon}^L_{ij}$ and $\bar{\varepsilon}^R_{ij}$.
    $Stack = (Stack \backslash \{\mathbf{Z}_i\}) \cup \{\mathbf{Z}^L_{ij}, \mathbf{Z}^R_{ij}\}$.
    $Stack_{out} = (Stack_{out} \backslash \{\mathbf{Z}_{O_i}\}) \cup \{\mathbf{Z}^L_{O_{ij}}, \mathbf{Z}^R_{O_{ij}}\}$.
end while
```

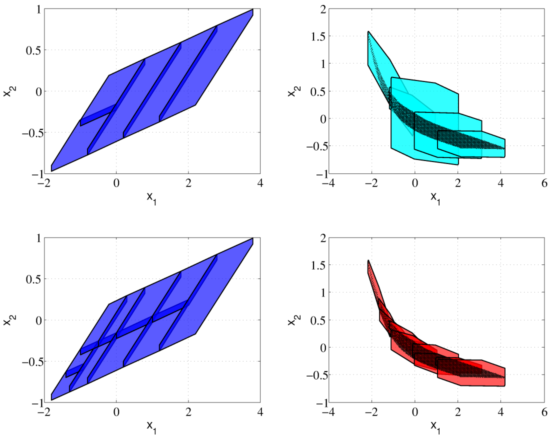

Fig. 13.13 In this example we considered the function f defined in (13.14). The starting sets $\mathbf{Z}$ are depicted on the left. In one case, $\mathbf{Z}$ has been split into 5 while in the other into 10. The splits have been done according to Algorithm 13.1. The results are depicted on the right, in cyan the output obtained with 5 splits while in red the one with 10 splits.

13.5 Capture Basin

Consider the following system

$$x_{i+1} = \begin{cases} f(x_i, \kappa_f(x)), & \forall x \in \mathcal{X}_F \\ \bar{f}_R(x_i), & \forall x \in R, R \nsubseteq \mathcal{X}_F \end{cases} \tag{13.10}$$

the system $x_{i+1} = f(x_i, u_i)$ in closed-loop with the approximated control law (13.7), for all sets $R \in \bar{R}$ under the dual mode approach. Given (13.10), we define with R_s the capture basin [4, 28], i.e., the set of initial states such that the terminal invariant

Algorithm 13.2 Computation of the capture basin

```
Require: Discrete time system (13.10) and $\overline{R}$, set of all hypercubic regions.
Ensure: Capture basin $R_s$.
Initialize $R_s = \mathcal{X}_F$ and $\Xi = \{R : R \subseteq \bar{R},\ R \nsubseteq R_s\}$.
while $\Xi \neq \emptyset$ do
    $X_{temp} = R_s$.
    for all $R \in \Xi$, compute an outer approximation $R_O(1)$ of the 1-step ahead reachable set by
    using Algorithm 13.1.
    Add to $X_{temp}$ all the boxes $R$ that satisfy $R_O(1) \subseteq R_s$. $X_{temp}$ represents an inner approximation
    of the 1-step backward reachable set of $R_s$.
    Set $R_s = X_{temp}$.
    $\Xi_{prec} = \Xi$.
    Update $\Xi$, as $\Xi = \{R : R \subseteq \bar{R},\ R \nsubseteq R_s\}$.
    if $\Xi_{prec} == \Xi$ then
        Return $R_s$.
    end if
end while
```

set $\mathcal{X}_F$ is reached in finite time while at the same time satisfying the state and the control constraints $\mathcal{X}$ and $\mathcal{U}$. Note that, since system constraints are satisfied in $\mathcal{X}_F$ and $\mathcal{X}_F$ is by definition invariant for the closed-loop system (13.10), one has $R_s \supseteq \mathcal{X}_F$.

The exact computation of the capture basin R_s is a difficult problem for the case of nonlinear systems. For this reason we suggest Algorithm 13.2 to compute an inner approximation of R_s. Algorithm 13.2 makes use of Algorithm 13.1. With Algorithm 13.2 we check which boxes R belong to R_s. Since the interpolated control law guarantees the satisfaction of the control constraints if $\mathcal{U}$ is convex (see [30]), and state constraint $\mathcal{X}$ are satisfied by requiring $\bar{R} \subseteq \mathcal{X}$, Algorithm 13.2 has just to check from which boxes the set $\mathcal{X}_F$ is attainable in finite time.

13.5.1 Approximate Explicit NMPC

In the following we introduce a recursive algorithm for multiresolution approximation of explicit nonlinear model predictive control laws. The algorithm is initialized with a user-defined coarse uniform grid before a dyadic refinement strategy is used to improve the approximation to a specified accuracy. Exploiting the fact that the state space can be decomposed into a union of hypercubes R (with respect to the approximate receding horizon control law), the algorithm restricts the dyadic refinement to the hypercubes intersecting the current invariant set. In this way the basin of attraction is constructed from the inside out (starting from the terminal set). The procedure is summarized in Algorithm 13.3. The algorithm requires the NMPC problem (13.1) and the NMPC cost function (13.2). The index set Λ is initialized at level l_0 along with all indices and details. The set of stored detail coefficients is given by the set $\mathbf{w}$. When the grid is refined, Λ stores the levels of resolution k and

Algorithm 13.3 Adaptive Hierarchical Approximate NMPC

```
Require: NMPC problem (13.1), NMPC Cost Function (13.2), $l_0$, and $l_{\max}$.
Ensure: detail coefficients $\mathbf{w}$ and index set $\Lambda$ such that the system $x_{i+1} = f(x_i, u_i)$ in closed-loop
    with the approximate control law $\hat{u}(x)$ (see (13.7)) has guaranteed feasibility and stability over
    the capture basin $R_s$.
Initialize the index set $\Lambda = \{(k,i) : i \in I_k, k = l_0\}$ and the initial set of hypercubes
Initialize the capture basin $R_s = \mathcal{X}_F$ and the set of intersecting hypercubes $R_c = \{R \in R_{\text{active}} :
    R \cap R_s \neq \emptyset\}$ where $R_{\text{active}}$ is the set of hypercubes not contained within $R_s$
Compute the initial details $\mathbf{w} = \{w_{k,i} : (k,i) \in \Lambda\}$ by solving the NMPC problem (13.1) point-
    wise at the vertices of all $R \in R_{\text{active}}$
while $R_c \neq \emptyset$ do
    Compute the capture basin $R_s$ with Algorithm 13.2.
    Recompute the set of candidate refinement hypercubes $R_c = \{R \in R_{\text{active}} : R \cap R_s \neq \emptyset,\ l_R \leq
        l_{\max}\}$ where $l_R$ is the level of the hypercube
    Refine all hypercubes $R \in R_c$
    Update $R_{\text{active}}$ and define the set of new vertices as $\Lambda_n$
    Solve the NMPC problem (13.1) at the new vertices and compute the new detail coefficients
        $\mathbf{w}_n$
    Update the index set $\Lambda = \Lambda \cup \Lambda_n$
    Update the detail set $\mathbf{w} = \mathbf{w} \cup \mathbf{w}_n$
end while
```

indices corresponding to the set of hierarchical details that are not discarded due to being initial conditions not feasible for problem (13.1). The maximum level of resolution is given as $l_{\max}$. The capture basin is computed using Algorithm 13.2. The set of hypercubes R_c intersecting R_s represents the set of refinement candidate sets. R_{active} is the set of hypercubes not contained within R_s; note that $R_c \subseteq R_{\text{active}}$. See [30] for details about the complexity of the real-time implementation of the approximate control.

The main theorem is now stated. It proves that Algorithm 13.3 always provides a stabilizing receding horizon control law and verifiable region of attraction for the NMPC problem (13.1). Note that we adopt a dual mode strategy, i.e., once the terminal set is attained, the stabilizing feedback law defined in $\mathcal{X}_F$ is applied.

Theorem 13.7. *Let $\hat{u}_0$ be the resulting receding horizon approximate control law computed from Algorithm 13.3 for the NMPC problem with cost (13.2), $l_0 \in \mathbb{N}$, and $l_{\max} \in \mathbb{N}$. The following properties hold for $\hat{u}_0$:*

a) Asymptotic stability to the origin for all $x_0 \in R_s$
b) $\hat{u}_0 \in \mathcal{U}$ for all $x \in R_s$
c) For all $x_0 \in R_s$, $x_i \in \mathcal{X}$ for all $i = 1, 2, 3, \ldots$
d) $R_s \supseteq \mathcal{X}_F$
e) As $l_{\max} \to \infty$, then $\hat{u}_0 \to u_0^*$ and $R_s \to \mathcal{R}$ where $\mathcal{R}$ is the maximum invariant set for (13.1).

See proof of Theorem 12 in [31].

13.6 Numerical Example

Consider the following two-dimensional continuous-time nonlinear system (e.g., see [10, 19, 9]):

$$\dot{x}_1(t) = x_2(t) + [0.5 + 0.5x_1(t)]u(t) \tag{13.11}$$

$$\dot{x}_2(t) = x_1(t) + [0.5 - 2x_2(t)]u(t) \tag{13.12}$$

It is well known (see [10]) that the origin of the system governed by (13.11) and (13.12) is unstable, and that the linearized system is stabilizable (but not controllable).

In consideration of the NMPC problem (13.1), the system (13.11) and (13.12) is discretized using a forward difference Euler approximation with sampling time $T = 0.1$. The input and state constraint sets are $\mathcal{U} = \{u \in \mathbb{R} : |u| \leq 2\}$ and $\mathcal{X} = \{x \in \mathbb{R}^2 : ||x||_\infty \leq 1\}$. The cost function is defined over a prediction horizon of length $N = 15$ as

$$J(u_0,\ldots,u_{N-1},x_0,\ldots,x_N) := x_N^T P x_N + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i$$

where

$$Q = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \quad R = 0.01, \quad P = \begin{bmatrix} 19.6415 & 13.1099 \\ 13.1099 & 19.6414 \end{bmatrix}.$$

The terminal penalty matrix P as well as the auxiliary controller $u = -Kx$, are computed using a linear differential inclusion (LDI, see [7]), in place of the original nonlinear system, and thus determine an invariant ellipsoid $\mathcal{X}_F = \{x \in \mathbb{R}^2 : x^T P x \leq 1\}$ for an uncertain linear time-varying system. With $l_0 = 2$ and $l_{\max} = 8$, we compute a stabilizing control law using Algorithm 13.3. In Table 13.2 we compare the results obtained by computing the capture basin with pure interval arithmetic (i.e., the splitting procedure has not been used) and Algorithm 13.1. As we can see, Algorithm 13.1 with 5 and 10 splits provides a capture basin that is sligthly bigger than the one obtained with interval arithmetic but, at the same time, the number of points describing the interpolated control law is 40% less. This motivates the use of Algorithm 13.1 instead of pure interval arithmetic. Note that, by the comparison to Algorithm 13.1 with 5 and 10 splits, we conclude that the use of more than 10 splits will not add further value since it will imply more computational effort for a very small improvement in the volume of the capture basin. It is important to note that the number of splits necessary to better describe the capture basin is problem dependent. In Figure 13.14 and Figure 13.15 the approximate receding horizon control law and an approximation of the capture basin, obtained with Algorithm 13.1 and 10 splits are shown.

ACADO [18] Toolkit has been used in order to solve the pointwise NMPC problem (13.1), while MPT toolbox [23] and the INTLAB interval toolbox [27] have been used to recursively compute the capture basin and the outer approximations.

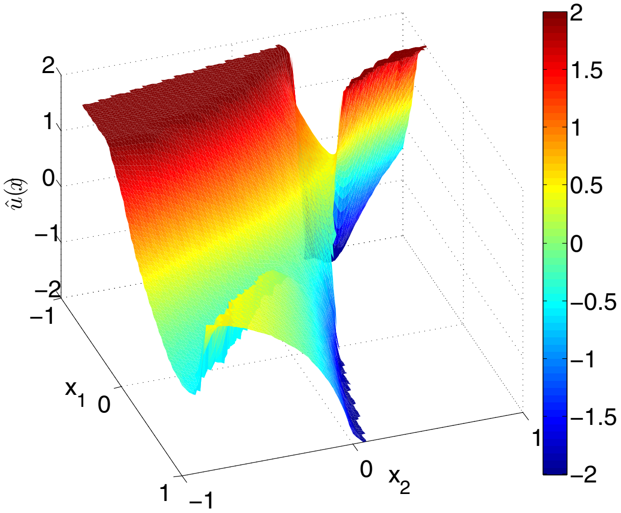

Fig. 13.14 Approximate control law $\hat{u}_0(x)$.

13.7 Conclusion

The approximate explicit NMPC method we have presented combines an adaptive hierarchical gridding scheme with a verification method based on reachability analysis. The approach approximates the optimal control law directly, and because of the basis functions used to build the function approximation, can be tuned in order to control the complexity and accuracy of the solution. This ability to guarantee a level of accuracy at the grid points enables an adaptive approach based on thresholding

Table 13.2 Comparison between pure interval arithmetic and Algorithm 13.1.

	# of points describing $\hat{u}_0(x)$	capture basin volume
Interval analysis	5808	2.5857
5 splits	3571	2.6035
10 splits	3439	2.5996

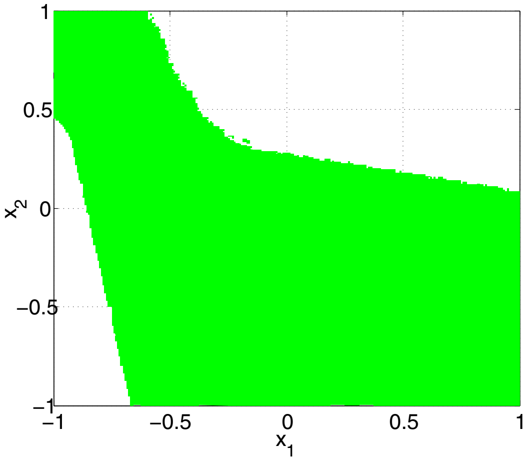

Fig. 13.15 Feasible and stable region.

that can lead to sparse representations of the explicit control law, while preserving guaranteed feasibility and stability of the solution. By employing reachability methods based on zonotopes and DC programming, the complexity of the function approximation and verification procedure is significantly decreased. A direct result of this reduction in complexity is a smaller storage requirement for the receding horizon control law and a larger verifiable region of attraction.

Appendix

13.7.1 Models Used in the Examples

The examples given in Sec. 13.4 are based on the following models.

Model 1:

$$\begin{cases} x_1(k+1) &= 3x_1(k) - \dfrac{x_1(k)^2}{7} - \dfrac{4x_1(k)x_2(k)}{4+x_1(k)} \\ x_2(k+1) &= -2x_2(k) + \dfrac{3x_1(k)x_2(k)}{4+x_1(k)} \end{cases} \tag{13.13}$$

Model 2:

$$\begin{cases} x_1(k+1) & = x_1(k) + 0.4x_2(k) \\ x_2(k+1) & = -0.132e^{(-x_1(k))}x_1(k) - 0.213x_1(k) + 0.274x_2(k) \end{cases} \tag{13.14}$$

Acknowledgements This research was partially supported by the Swiss National Science Foundation under grant 200021-122072 and by the European Commission under the projects Feednetback FP7-ICT-223866 [15] and EMBOCON FP7-ICT-2009-4 248940 [14].

References

1. Adjiman, C., Floudas, C.: Rigorous convex underestimators for general twice-differentiable problems. Journal of Global Optimization **9**(1), 23–40 (1996)
2. Alamo, T., Bravo, J., Camacho, E.F.: Guaranteed state estimation by zonotopes. Automatica **41**(6), 1035–1043 (2005)
3. Alamo, T., Bravo, J., Redondo, M., Camacho, E.: A set-membership state estimation algorithm based on DC programming. Automatica **44**(1), 216–224 (2008)
4. Aubin, J.: Viability Theory. Birkhauser Boston Inc., Cambridge, MA, USA (1991)
5. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. Automatica **38**(1), 3–20 (2002)
6. Berz, M., Makino, K.: Suppression of the wrapping effect by Taylor model-based verified integrators: Long-term stabilization by shrink wrapping. International Journal of Differential Equations and Applications **10**(4), 385–403 (2005)
7. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear matrix inequalities in system and control theory, vol. 15. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1994)
8. Canale, M., Fagiano, L., Milanese, M.: Set membership approximation theory for fast implementation of model predictive control laws. Automatica **45**(1) (2009)
9. Canale, M., Fagiano, L., Milanese, M., Novara, C.: Set membership approximations of predictive control laws: The tradeoff between accuracy and complexity. In: European Control Conference (ECC2009). Budapest, Hungary (2009)
10. Chen, H., Allgöwer, F.: A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica **34**(10), 1205–1218 (1998)
11. Combastel, C.: A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes. In: Proc. 44th IEEE Conf. Decision and Control (CDC2005), and 2005 European Control Conf. (CDC-ECC'05), pp. 7228–7234. Seville, Spain (2005)
12. Delanoue, N., Jaulin, L., Hardouin, L., Lhommeau, M.: Guaranteed characterization of capture basins of nonlinear state-space systems. In: Informatics in Control, Automation and Robotics: Selected Papers from the International Conference on Informatics in Control, Automation and Robotics 2007 (ICINCO2007), pp. 265–272. Springer, Angers, France (2008)
13. Donzé, A., Clermont, G., Legay, A., Langmead, C.: Parameter synthesis in nonlinear dynamical systems: Application to systems biology. In: Proc. 13th Annual Int. Conf. Research in Computational Molecular Biology, pp. 155–169 (2009)
14. EMBOCON www.embocon.org
15. Feednetback www.feednetback.eu
16. Fukuda, K.: From the zonotope construction to the Minkowski addition of convex polytopes. Journal of Symbolic Computation **38**(4), 1261–1272 (2004)
17. Horst, R., Thoai, N.: DC programming: Overview. Journal of Optimization Theory and Applications **103**(1), 1–43 (1999)
18. Houska, B., Ferreau, H., Diehl, M.: ACADO toolkit—An open-source framework for automatic control and dynamic optimization. Optimal Control Applications and Methods **32**(3), 298–312 (2011)

19. Johansen, T.A.: Approximate explicit receding horizon control of constrained nonlinear systems. Automatica **40**(2), 293–300 (2004)
20. Johansen, T.A., Petersen, I., Slupphaug, O.: On explicit suboptimal LQR with state and input constraints. In: Proc. 39th IEEE Conf. Decision and Control (CDC2000), pp. 662–667. Sydney, Australia (2000)
21. Kearfott, R.: Rigorous global search: Continuous problems, vol. 13. Kluwer Academic Publishers, Dordrecht-Boston-London (1996)
22. Kuhn, W.: Rigorously computed orbits of dynamical systems without the wrapping effect. Computing **61**(1), 47–67 (1998)
23. Kvasnica, M., Grieder, P., Baotić, M.: Multi-Parametric Toolbox (MPT) (2004). URL http://control.ee.ethz.ch/ mpt/
24. Makino, K.: Rigorous analysis of nonlinear motion in particle accelerators. Ph.D. thesis, Michigan State University (1998)
25. Moore, R.E.: Interval Analysis, vol. 60. Prentice Hall, Englewood Cliffs, NJ (1966)
26. Pin, G., Filippo, M., Pellegrino, A., Parisini, T.: Approximate off-line receding horizon control of constrained nonlinear discrete-time systems. In: Proc. European Control Conference, pp. 2420–2431. Budapest, Hungary (2009)
27. Rump, S.: INTLAB - INTerval LABoratory. In: T. Csendes (ed.) Developments in Reliable Computing, pp. 77–104. Kluwer Academic Publishers, Dordrecht (1999). http://www.ti3.tu-harburg.de/rump/
28. Saint-Pierre, P.: Approximation of the viability kernel. Applied Mathematics and Optimization **29**(2), 187–109 (1994)
29. Seron, M.M., Goodwin, G.C., Dona, J.A.D.: Geometry of model predictive control including bounded and stochastic disturbances under state and input constraints. Tech. rep., University of Newcastle, Newcastle, Australia (2000)
30. Summers, S., Jones, C.N., Lygeros, J., Morari, M.: A multiscale approximation scheme for explicit model predictive control with stability, feasibility, and performance guarantees. In: Proc. IEEE Conf. Decision and Control (CDC2009), pp. 6328–6332. Shanghai, China (2009)
31. Summers, S., Raimondo, D.M., Jones, C.N., Lygeros, J., Morari, M.: Fast explicit nonlinear model predictive control via multiresolution function approximation with guaranteed stability. In: Proc. 8th IFAC Symp. Nonlinear Control Systems (NOLCOS 2010). Bologna, Italy (2010). URL http://control.ee.ethz.ch/index.cgi?page=publications;action=details;id=3557
32. Tuy, H.: DC optimization: theory, methods and algorithms. Handbook of global optimization pp. 149–216 (1995)
33. Wan, J., Vehi, J., Luo, N.: A numerical approach to design control invariant sets for constrained nonlinear discrete-time systems with guaranteed optimality. Journal of Global Optimization **44**(3), 395–407 (2009)