

# On the computation of invariant sets for constrained nonlinear systems: An Interval Arithmetic Approach <sup>★</sup>

J.M. Bravo <sup>a</sup>, D. Limon <sup>b</sup>, T. Alamo <sup>b</sup>, E.F. Camacho <sup>b</sup>

<sup>a</sup>*Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática. Universidad de Huelva. Carretera Huelva - La Rábida. Palos de la Frontera, 21071 Huelva. Spain.*

<sup>b</sup>*Departamento de Ingeniería de Sistemas y Automática. Universidad de Sevilla. Avda Camino de los Descubrimientos s/n. 41092 Sevilla. Spain.*

---

## Abstract

This paper deals with the computation of control invariant sets for constrained nonlinear systems. The proposed approach is based on the computation of an inner approximation of the one step set, that is, the set of states that can be steered to a given target set by an admissible control action. Based on this procedure, control invariant sets can be computed by recursion.

We present a method for the computation of the one-step set using interval arithmetic. The proposed specialized branch and bound algorithm provides an inner approximation with a given bound of the error; this makes it possible to achieve a trade off between accuracy of the computed set and computational burden. Furthermore an algorithm to approximate the one step set by an inner bounded polyhedron is also presented; this allows us to relax the complexity of the obtained set, and to make easier the recursion and storage of the sets.

*Key words:* Nonlinear systems, Invariance, Constraints, Intervals.

---

## 1 Introduction

Set invariance theory is very important for the analysis of constrained autonomous systems and for the design of stabilizing controllers of constrained systems. The underlying theory of this problem is well established, and a large number of results have been presented since the seminal paper (Bertsekas & Rhodes 1971). In (Blanchini 1999), the most relevant results on this topic are summarized. The most general way of computing an invariant set is based on calculation of the so-called one step set. This is the set of the states that can be steered to a given target set by an admissible control action. Using this set in a recursive way, control invariant sets can be computed (Bertsekas & Rhodes 1971). The computation of the one step set can be considered as a geometric

problem; however, there is no general method to compute it. Algorithms exist for several kind of systems as for example, for linear systems (Gilbert & Tan 1991), for polytopic systems (Blanchini 1994) or for piece-wise affine systems (Kerrigan 2000). However, the computation of invariant sets for general nonlinear systems subject to constraints is an open field.

Invariance set theory has become a relevant topic in the MPC paradigm. As it is well known, one of the key ingredients for the stabilizing design of the MPC is an invariant terminal set (either positively or control invariant set) (Chen & Allgöwer 1998, Mayne 2001). The enlargement of this set provides a larger domain of attraction of the controller, which has motivated recent results on invariant sets. In (Chen, Ballance & O'Reilly 2001) the terminal set is enlarged by using a locally linear differential inclusion (LDI) representation for the nonlinear system and solving off-line an LMI optimization problem. In (Cannon, Deshmukh & Kouvaritakis 2003), a locally LDI representation is also used, and a polytopic terminal set and an associated terminal cost is computed. This result has been improved in (Cannon, Kouvaritakis & Deshmukh 2003). In (Limon, Alamo & Camacho 2004) the enlargement is achieved by replacing the invariant

---

<sup>★</sup> This paper was not presented at any IFAC meeting. Corresponding author: J.M Bravo. Tel. +034-959017367.

The authors would like to acknowledge MCYT-Spain (contracts DPI2002-04375-c03-01 and DPI2000-0666-c02-02) for funding this work and also to thank the anonymous reviewers for their helpful comments.

*Email addresses:* caro@uhu.es (J.M. Bravo), limon@cartuja.us.es (D. Limon), alamo@cartuja.us.es (T. Alamo), eduardo@cartuja.us.es (E.F. Camacho).

terminal set by a contractive sequence of sets. The computation of invariant sets can be also used for the design of time-optimal control laws (Mayne & Schröder 1997) or used as stabilizing constraint (Bemporad 1998, Limon, Alamo & Camacho 2003). In (Johansen 2004) an approximate explicit description of the MPC controller for constrained nonlinear systems is presented and the domain of attraction is also obtained.

In this paper, a new algorithm to compute an inner approximation to the one-step set for constrained nonlinear systems is presented. This is a specialized branch and bound algorithm based on interval arithmetic. This algorithm provides an inner approximation with a given error bound. This bound allows us to achieve a trade-off between accuracy and computational burden. To reduce the complexity of the obtained region, a second algorithm is proposed to compute an inner polytopic approximation of the obtained region. It is worth remarking that the proposed algorithm can be easily adapted to the structure of the system as well as extended to the robust case.

## 2 Definitions and problem statement

Consider a system described by a nonlinear discrete model:

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

where  $x_k \in \mathbb{R}^n$  is the system state and  $u_k \in \mathbb{R}^m$  is the control signal at sample time  $k$ . The system can be subject to control and state constraints:

$$x_k \in X, u_k \in U \quad (2)$$

where  $X$  and  $U$  are compact sets, both of them containing the origin. The model function  $f(\cdot, \cdot)$  is assumed to be continuous in  $X \times U$ .

A set  $\Omega \subset \mathbb{R}^n$  is a control invariant set for system (1) subject to constraint  $u \in U$ , if for all  $x \in \Omega$ , there exists an admissible input  $u \in U$  such that  $f(x, u) \in \Omega$ . To check or compute a control invariance set, the one-step set operator can be used (Bertsekas & Rhodes 1971). The one-step set of a given target set  $\Omega$ ,  $Q(\Omega)$ , for system (1) subject to  $u \in U$ , is the set of states which can be steered in one step to the target set  $\Omega$  by an admissible control action, i.e.  $Q(\Omega) = \{x \in \mathbb{R}^n : \exists u \in U \text{ such that } f(x, u) \in \Omega\}$ . Thus, the set  $\Omega$  is a control invariant set if and only if  $\Omega \subseteq Q(\Omega)$ . Given an admissible invariant set  $\Omega$  (which can be obtained, for instance, from the linearized system at the origin), the  $i$ -step stabilizable set can be obtained by the recursion  $X_i = Q(X_{i-1}) \cap X$ , where  $X_0 = \Omega$ . Note that every set  $X_i$  is a control invariant set and  $X_i$  is contained in the maximal stabilizable set  $X_\infty$ .

Procedures to compute the one-step set have been presented for linear systems subject to polytopic constraints (Blanchini 1999, Kerrigan 2000, Mayne 2001), for systems described by linear differential inclusions (Blanchini 1994) or for piece-wise affine systems (Kerrigan 2000). However, there is no general procedure to compute it for constrained nonlinear systems.

In order to relax the complexity of the computation, the one-step set  $Q(\Omega)$  can be replaced by an inner approximation  $Q_{ap}(\Omega)$  such that  $Q_{ap}(\Omega) \subseteq Q(\Omega)$ . This relaxation makes sense for the sake of tractability of the procedure used to compute it. Thus, the sets obtained by the recursion  $\Phi_i = Q_{ap}(\Phi_{i-1}) \cap X$  with  $\Phi_0 = \Omega$  can be considered as an estimation of the  $i$ -step stabilizable set since  $\Phi_i \subseteq X_i$ ; however  $\Phi_i$  may not be a control invariant set. Note that the set  $\Omega_i = \bigcup_{j=0}^i \Phi_j$  is a control invariant set and  $\Phi_i \subseteq \Omega_i \subseteq X_i$ .

These results show the relevance of designing algorithms computationally less demanding for the computation of an inner approximation of the one-step set, which is the objective of this paper. In the following sections it is shown how interval arithmetic can be used for this aim.

## 3 Interval arithmetic

An interval number  $X = [a, b]$  is the set  $\{x : a \leq x \leq b\}$ . Interval arithmetic is an arithmetic defined on sets of intervals, rather than sets of real numbers. The interval arithmetic is based on operations applied to sets of intervals. Let  $\mathbb{I}$  be the set of real compact intervals  $[a, b]$  with  $a, b \in \mathbb{R}$ . Operations in  $\mathbb{I}$  satisfy the expression:

$$A \text{ op } B = \{a \text{ op } b : a \in A, b \in B\} \text{ for } A, B \in \mathbb{I} \quad (3)$$

In this way, basic interval operations (Moore 1966) are calculated by simple floating point operations on their extremes, as for instance:

$$[a, b] + [c, d] = [a + c, b + d] \quad (4)$$

The interval extension of standard functions  $\{\sin, \cos, \tan, \arctan, \exp, \ln, \text{abs}, \text{sqr}, \text{sqrt}\}$  is possible too (Moore 1966). There are several interval software libraries that implement these operations (Knueppel 1994), as for instance the INTLAB interval toolbox (Rump 1999), which has been used in the example of this work. Note that the interval operations are implemented by a set of floating point operations over the extremes of the intervals.

In what follows, some basic definitions and well-known results on interval arithmetic are given. A box is an interval vector (vector whose components are intervals). A unitary box, denoted  $\mathbf{B}^m$ , is a box that consists of  $m$  unitary intervals  $[-1, 1]$ . An interval hull of a set  $X \subseteq \mathbb{R}^n$ ,

denoted by  $\square X$ , is a box that satisfies  $X \subseteq \square X$ . Given a box  $H = ([a_1, b_1], \dots, [a_n, b_n])^\top$ ,  $\text{mid}(H)$  denotes its center and  $\text{diam}(H) = (b_1 - a_1, \dots, b_n - a_n)^\top$ . The range of a continuous function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^s$  over a set  $X \subseteq \mathbb{R}^n$  is defined as  $f(X) = \{f(x) : x \in X\}$ .

**Definition 1 (Natural interval extension)** (Kearfott 1996) *If  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a function computable as an expression, algorithm or computer program involving the four elementary arithmetic operations interspersed with evaluations of standard functions then, a natural interval extension of  $f$ , denoted  $\square f$ , is obtained replacing each occurrence of each variable by the corresponding interval variable, by executing all operations according to interval operators and by computing ranges of the standard functions.*

**Theorem 2** (Moore 1966) *A natural interval extension  $\square f$  of a continuous function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  over a box  $X \subseteq \mathbb{R}^n$  satisfies that  $f(X) \subseteq \square f(X)$ .*

**Theorem 3** (Kearfott 1996) *A natural interval extension  $\square f$  of a continuous function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  over two boxes  $X, Y \subseteq \mathbb{R}^n$  such that  $X \subseteq Y$  satisfies that  $\square f(X) \subseteq \square f(Y)$ .*

#### 4 One-step set approximation algorithm

Interval algorithms have been used successfully in the resolution of several kind of problems. These algorithms are basically branch and bound algorithms where the ranges of functions are bounded by interval arithmetic (Hansen 1992). For instance, given two sets  $A$  and  $B$  defined by finite sets of inequalities, interval branch and bounds algorithms can prove set inclusion  $A \subset B$  and solve the set inversion problem  $A = f^{-1}(B)$  for a given continuous function (Jaulin, Kieffer, Didrit & Walter 2001).

In this paper, and assuming that sets  $X$  and  $U$  are boxes (or lists of boxes), an interval branch and bound algorithm to compute an inner approximation to the one-step set is presented. The input parameters of the proposed algorithm (Algorithm 1) are the nonlinear system (1), the sets of admissible states  $X$  and inputs  $U$ , the target set  $\Omega$  and a tolerance  $\varepsilon$ , that bounds the error of the approximation. The algorithm returns the one-step set approximation  $B(\Omega)$ .

The algorithm initializes a list  $L$  with the box<sup>1</sup>  $Z = X \times U \subseteq \mathbb{R}^{n+m}$  and while the list is not empty, the algorithm processes boxes that obtains from  $L$ . At the iteration  $i$ , the algorithm extracts the greatest box from

<sup>1</sup> If the problem is not constrained in the states or in the inputs, then an artificial sufficiently large box can be considered as constraint.

$L$ , denoted  $Z_i = (X_i, U_i)$ , and computes a control action  $u_i = \text{select}(Z_i)$ . The operator  $\text{select}()$  is a design knob of the algorithm; it can be chosen for instance as  $\text{select}(Z_i) = \text{mid}(U_i)$  or computing the 'best' control input  $u_i$  such that  $X_i$  is driven to set  $\Omega$ ; for example, if the system is affine in  $u_k$ , that is  $f(x_k, u_k) = g(x_k) + h(x_k)u_k$ ,  $g(x_k)$  and  $h(x_k)$  can be approximated by LDIs and  $\Omega$  is represented by a set of linear constraints, then it is possible to formulate a linear program to obtain a feasible  $u_i$  such that  $f(X_i, u_i) \subseteq \Omega$  (Boyd, Ghaoui, Feron & Balakrishnan 1994, Cannon, Deshmukh & Kouvaritakis 2003).

Once the control action  $u_i$  is obtained, three actions are possible on  $Z_i = (X_i, U_i)$ . If the interval evaluation of  $\square f(X_i, U_i)$  fulfills the condition  $\square f(X_i, U_i) \cap \Omega = \emptyset$  then the box is rejected. If the interval extension  $\square f(X_i, u_i)$  belongs to  $\Omega$  then the box  $X_i$  is inserted in  $B(\Omega)$  and all boxes  $Z_j = (X_j, U_j) \subseteq L$  such that  $X_j = X_i$  are erased from the list. In other case, the width of  $Z_i$  is checked and if its width is smaller than  $\varepsilon$  then the box is discarded. If its width is bigger than  $\varepsilon$  then the box is split into two new boxes  $Z_1, Z_2$  such that  $Z_i = Z_1 \cup Z_2$ . Taking into account the size of  $X_i$  and  $U_i$ , the algorithm decides which component of  $Z_i = (X_i, U_i)$  must be split. If the component  $X_i$  is bisected then it is necessary to bisect all boxes  $Z_j = (X_j, U_j) \subseteq L$  such that  $X_j = X_i$ . This allows us to erase non necessary boxes of  $L$  when a box is inserted in  $B(\Omega)$ .

#### Algorithm 1

$B(\Omega) = \text{OneStepSetApproximation}(\Omega, f, X, U, \varepsilon)$

**Alg**

$L = (X, U)$

**while**  $L \neq \emptyset$

$Z_i = \text{greatest box in } L$ ;

$X_F = \square f(X_i, U_i)$ ;

$X_S = \square f(X_i, \text{select}(Z_i))$

$\delta_{X_i} = \|\text{diam}(X_i)\|_\infty$ ,  $\delta_{U_i} = \|\text{diam}(U_i)\|_\infty$

$\delta = \max\{\delta_{X_i}, \delta_{U_i}\}$

**if**  $(X_F \cap \Omega \neq \emptyset)$  **action** = BISECT **endif**

**if**  $(X_F \cap \Omega = \emptyset)$  **or**  $(\delta < \varepsilon)$

**action** = DISCARD **endif**

**if**  $(X_S \subseteq \Omega)$  **action** = INSERT **endif**

**if** (action==DISCARD) erase  $Z_i$  from  $L$  **endif**

**if** (action==INSERT)

insert  $X_i$  in  $B(\Omega)$

Erase from  $L$  all boxes  $Z_j = (X_j, U_j)$  such that  $X_j = X_i$

**endif**

**if** (action==BISECT)

**if**  $\delta_{X_i} > \delta_{U_i}$

**forall**  $Z_j \in L$  such that  $X_j = X_i$

$(X_{j_1}, X_{j_2}) = \text{bisect}(X_j)$

insert the boxes

$(X_{j_1}, U_j), (X_{j_2}, U_j)$  in  $L$

**endifor**

```

else  $(U_{i_1}, U_{i_2}) = \text{bisect}(U_i)$  and
      insert the boxes  $(X_i, U_{i_1}), (X_i, U_{i_2})$  in  $L$ 
endif
endif
endwhile
return  $B(\Omega)$ 
End

```

**Remark 1** For a given candidate box, the proposed algorithm can classify it into one of the following three cases: if the algorithm determines that it is contained in  $Q(\Omega)$ , then it is inserted in  $B(\Omega)$ ; if the algorithm concludes that it is contained in  $X \setminus Q(\Omega)$  or its size is smaller than the tolerance  $\varepsilon$ , then it is rejected. In any other case, the box is split. Therefore, the size of all the boxes of  $B(\Omega)$  is greater than or equal to  $\varepsilon$  and all the discarded boxes are such that either they are not contained in  $Q(\Omega)$  or they are smaller than  $\varepsilon$ .

In the following theorem the correctness and convergence of Algorithm 1 is proved.

**Theorem 4** Let  $\Omega, X, U$  be compact sets, let  $\check{\Omega}$  be the interior of  $\Omega$  and consider a system (1) where  $f(\cdot, \cdot)$  is continuous in  $\Omega \times U$ . Consider  $B(\Omega)$  obtained by the proposed algorithm.

- (1) For all  $x \in B(\Omega)$  there exists a control action  $u \in U$  such that  $f(x, u) \in \Omega$ . That is,  $B(\Omega) \subseteq Q(\Omega)$ .
- (2) For all  $x$  that can be steered to  $\check{\Omega}$  in an admissible way, there is a real positive number  $\varepsilon > 0$  such that  $x \in B(\Omega)$ .

**Proof:**

Suppose that  $x \in B(\Omega)$ , then there is a box  $X_i \subseteq B(\Omega)$  and  $u_i \in U$  such that  $x \in X_i$  and  $\square f(X_i, u_i) \subseteq \Omega$ . From Theorem 2 it is then inferred that  $f(x, u_i) \in f(X_i, u_i) \subseteq \square f(X_i, u_i) \subseteq \Omega$ . Thus  $B(\Omega) \subseteq Q(\Omega)$ .

To prove the second part is considered that  $z$  denotes the vector  $(x, u)$  and  $Ball(z, r) = \{\omega \in \mathbb{R}^{n+m} : \|\omega - z\|_\infty \leq r\}$  denotes the ball of radius  $r$  and center  $z$ . Consider any  $x \in X$  such that there is a  $u \in U$  satisfying  $f(x, u) \in \check{\Omega}$ , i.e.  $x \in Q(\check{\Omega}) \cap X$ , then by continuity of  $f(\cdot, \cdot)$  there exists a  $Ball(z, r)$  with  $z = (x, u)$ , such that  $f(Ball(z, r)) \subseteq \check{\Omega}$ . By interval monotonic inclusion (Theorem 3), there exists a box  $Z$  with  $z = \text{mid}(Z)$  and contained in  $Ball(z, r)$  such that  $\square f(Z) \subseteq f(Ball(z, r)) \subseteq \check{\Omega}$ . Denote  $\varepsilon = 0.5 \cdot \|\text{diam}(Z)\|_\infty$ . It is clear that any box  $\hat{Z}$  containing  $z$  and of size  $\|\text{diam}(\hat{Z})\|_\infty \leq \varepsilon$  is contained in  $Z$ . Therefore, by theorem 3,  $\square f(\hat{Z}) \subseteq \square f(Z) \subseteq \check{\Omega}$ . Then, any box  $\hat{Z}$  fulfills  $\square f(\hat{Z}) \subseteq \check{\Omega}$ , and hence  $\hat{Z} \subseteq Q(\Omega)$ .

Therefore, if Algorithm 1 is executed with an error bound  $\varepsilon$ , then any box  $\hat{Z}$  of size smaller or equal to  $\varepsilon$  containing  $(x, u)$  is not discarded. Consequently, this box is inserted in  $B(\Omega)$  by Algorithm 1. ■

A first consequence of Theorem 4 is that if  $\varepsilon \rightarrow 0$  then  $B(\Omega) \rightarrow Q(\Omega)$ , so  $B(\Omega)$  is a reliable approximation of the exact set  $Q(\Omega)$ . This provides an approximation with a given bound of the error, which allows us to choose a trade off between accuracy of the computed set and computational burden. Because  $X, U$  are the search space of Algorithm 1, state and control constraints are fulfilled. Finally, notice that the proposed algorithm can be easily extended to the robust case.

**Remark 2** In general, the cost of the computation of the one-step set for constrained nonlinear systems increases exponentially with the order of the system. A measure of the complexity of the algorithm is the worst case number of generated boxes from the initial box. If  $\varepsilon$  is the tolerance of the algorithm and  $n$  is the dimension of state vector, then the maximum number of generated boxes is proportional to  $(\frac{1}{\varepsilon})^n$ . The proposed algorithm reduces dramatically this worst case estimation of number of boxes, as it is shown in the example.

The interval algorithm proposed in this paper allows one to adapt the algorithm to the considered system by means of the operator  $\text{select}(\cdot)$ ; a good choice of this operator would provide a reduction on the computational burden. Finally notice that the computation of the one-step set is executed off-line, and therefore the computational cost is not a critical issue.

**Remark 3** The proposed algorithm requires a discrete-time model of the system; if the model is obtained from a continuous time model using some integration method (such as a Runge-Kutta, for instance), then integration errors may appear. The proposed algorithm is reliable, in the sense that all the boxes of  $B(\Omega)$  are contained in  $Q(\Omega)$ , assuming that the discrete-time model is exact. If the integration method is badly tuned, the integration error might be significant and the reliability of the obtained approximation of the one-step set might be reduced. This problem can be overcome using integration methods with guaranteed estimation (Moore 1966), which consider the integration error in the computation of the evolution of the state at the next sample time.

## 5 One-step set polytopic approximation

The approximate one-step set obtained from Algorithm 1 is represented by a list of boxes. In order to obtain a simpler representation, polytopes are proposed. Polytopes provide reliable approximations of convex regions and require less memory resources than lists of boxes. The possible non-convex nature of the obtained one-step set  $B(\Omega)$  leads to an increased conservativeness when an inner polytope approach is adopted. However, its simple and compact representation makes it suitable for the synthesis of controllers, as for instance those based on stabilizing constraints (Limon et al. 2004, Limon et al. 2003).

In this section we present Algorithm 2 that approximates the list of boxes  $B(\Omega)$  by an inner polytope. First, a preliminary definition is given.

**Definition 1 ( $\alpha$ -Support Hyperplane)** *Given a positive constant  $\alpha \in \mathbb{R}$ , a box  $H_i$  and a list of boxes  $B(\Omega)$ , an  $\alpha$ -support hyperplane is an hyperplane  $c^t x = 1$  such that:*

- $c^t x > 1$  for every  $x \in H_i$ .
- $\alpha c^t x \leq 1$  for every  $x \in B(\Omega)$ .

A box  $H_i$  can be written as  $H_i = \{x \in \mathbb{R}^n : \exists b \in \mathbf{B}^n \text{ such that } x = p_{H_i} + M_{H_i} b\}$ , where  $p_{H_i} \in \mathbb{R}^n$  and  $M_{H_i} \in \mathbb{R}^{n \times n}$  is a diagonal matrix. Thus, any box  $G_j$  of the list  $B(\Omega) = \bigcup_{j=1}^m G_j$  can be represented by  $p_{G_j} \in \mathbb{R}^n$  and the diagonal matrix  $M_{G_j} \in \mathbb{R}^{n \times n}$ . To maximize the value of  $\alpha$  of the  $\alpha$ -support hyperplanes, the optimization problem  $\Pi(B(\Omega), H_i)$  can be posed:

$$\begin{aligned} \min_{\gamma, c} \quad & \gamma \\ \text{s.t.} \quad & c^t p_{H_i} - \|c^t M_{H_i}\|_1 > 1 \\ & c^t p_{G_j} + \|c^t M_{G_j}\|_1 \leq \gamma \quad j = 1, \dots, m \\ & \gamma > 0 \end{aligned}$$

where  $\gamma = \alpha^{-1}$ . This optimization problem is based in the fact that  $c^t(p + Mb) < 1$  for all  $b \in \mathbf{B}^n$  if and only if  $c^t p + \|c^t M\|_1 < 1$ . The proposed optimization problem can be recast as a linear one and it can be solved with specialized algorithms.

Now, Algorithm 2 is presented. Given a target set  $\Omega$  and set  $B(\Omega)$  obtained by Algorithm 1, Algorithm 2 returns a polytope denoted  $P(\Omega)$  such that  $P(\Omega) \subseteq B(\Omega) \subseteq Q(\Omega)$ . Hence  $P(\Omega)$  can also be used as an approximation to the one-step set.

Suppose that  $B^c(\Omega) = X \setminus B(\Omega)$  is a list of boxes such that  $B^c(\Omega) = \bigcup_{i=1}^l H_i$ . Set  $B^c(\Omega)$  can be computed using Algorithm 1 with a minor modification<sup>2</sup> or by a simple division algorithm similar to this. The division algorithm provides boxes  $H_i \subseteq X$  and checks if those boxes do not belong to the obtained set  $B(\Omega)$ .

Considering as input parameters  $B(\Omega)$  and the complementary set  $B^c(\Omega)$ , Algorithm 2 returns an inner polytope  $P(\Omega)$ .

### Algorithm 2

$$P(\Omega) = \text{InsidePolytope}(\Omega)$$

<sup>2</sup> Algorithm 1 can be easily modified to compute both  $B(\Omega)$  and  $B^c(\Omega)$ : when action is equal to DISCARD, if the erased box is  $Z_i = (X_i, U_i)$ , then  $X_i$  would be included in  $B^c(\Omega)$  if  $X_i \cap X_j = \emptyset$  for all  $Z_j = (X_j, U_j) \in L$ .

### Alg

```

Compute  $B(\Omega)$  and  $B^c(\Omega)$ 
 $P(\Omega) = \square B(\Omega)$ 
for all  $H_i \in B^c(\Omega)$ 
  if  $H_i \cap P(\Omega) \neq \emptyset$ 
     $S_i = \{x \in \mathbb{R}^n : c_i^t x \leq 1\}$  where  $(c_i, \gamma_i)$ 
      are obtained from problem  $\Pi(B(\Omega), H_i)$ 
     $P(\Omega) = P(\Omega) \cap S_i$ 
     $P(\Omega) = \text{Delete redundant constraints to } P(\Omega)$ 
  endif
endifor
return  $P(\Omega)$ 

```

### End

**Theorem 5** *Let  $B(\Omega)$  be a set represented by a list of boxes, and  $P(\Omega)$  a polytope calculated by Algorithm 2 then  $P(\Omega) \subseteq B(\Omega)$ .*

**Proof:** Algorithm 2 returns a polytope  $P(\Omega) = \bigcap_{i=1}^p S_i$  where  $S_i = \{x : c_i^t x \leq 1\}$ . If  $x \in P(\Omega)$  then  $c_i^t x \leq 1, i = 1, \dots, p$ . For every  $H_i \in B^c(\Omega)$  there is an  $i \in [1, \dots, p]$  such that  $c_i^t x > 1 \forall x \in H_i$ . So,  $x \in B^c(\Omega)$  implies that  $x \notin P(\Omega)$ . This means that  $P(\Omega) \cap B^c(\Omega) = \emptyset$  and therefore  $P(\Omega) \subseteq B(\Omega)$ . ■

## 6 Example

As illustrative example, the technique is applied to a system used in [5] and described by the following ODEs:

$$\begin{aligned} \dot{x}_1 &= x_2 + u(\mu + (1 - \mu)x_1) \\ \dot{x}_2 &= x_1 + u(\mu + 4(1 - \mu)x_2) \end{aligned}$$

where the parameter  $\mu$  is equal to 0.5 and the input constraint is  $|u| \leq 2$  and the states are constrained to  $\|x\|_\infty \leq 1$ . The system is discretized with a sampling time of 0.1 time-units using a fourth order Runge-Kutta method. A linear locally stabilizing state feedback gain  $K = [2.118 \ 2.118]$  is used to derive the initial control invariant set  $\Omega = \{x \in \mathbb{R}^2 : 16.5926 x_1^2 + 23.1854 x_1 x_2 + 16.5926 x_2^2 \leq 0.7\}$  (Chen & Allgöwer 1998).

Figure 1 shows the one step approximation  $B(\Omega)$  computed using Algorithm 1 for two cases: with  $\varepsilon = 0.03$  and  $\varepsilon = 0.01$ , which provides a better approximation to the one step set. In the case of  $\varepsilon = 0.01$ ,  $B(\Omega)$  and  $B^c(\Omega)$  are compounded by 315 and 2117 boxes respectively. The smaller box has a diameter of 0.0078. To obtain a similar result with a grid method 65536 points are needed. So, 2694% more evaluations of the discrete-time system to obtain the same result are required.

Figure 2 shows a sequence  $\Phi_i$  with  $i = 1, \dots, 10$  where  $\Phi_i = B(\Phi_{i-1})$ ,  $\Phi_0 = \Omega$  and using Algorithm 1 with  $\varepsilon = 0.01$ . A sequence  $P_i$  with  $i = 0, \dots, 9$ ,  $P_i = P(\Phi_{i-1})$  and  $P_0 = P(\Omega)$  is obtained by applying Algorithm 2. This sequence can be seen in Figure 3.

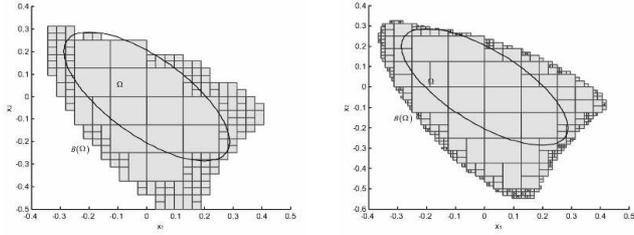


Fig. 1. The set  $B(\Omega)$  for  $\varepsilon = 0.03$  and  $\varepsilon = 0.01$

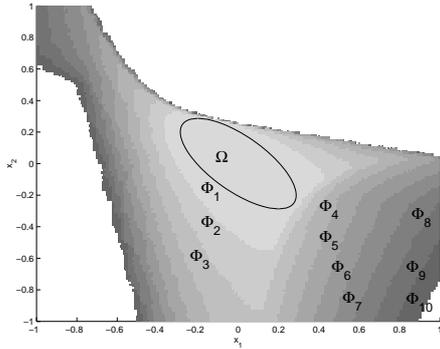


Fig. 2.  $\Phi_i$  with  $i = 1..10$  where  $\Phi_i = B(\Phi_{i-1})$  and  $\Phi_0 = \Omega$

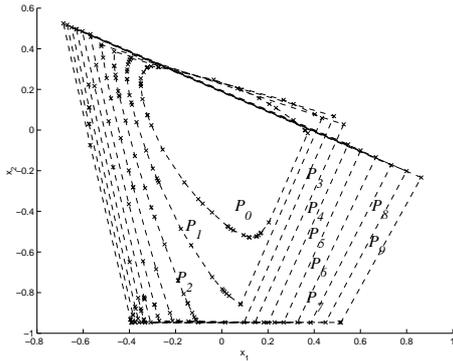


Fig. 3.  $P_i$  with  $i = 0..9$ ,  $P_i = P(P_{i-1})$  and  $P_0 = P(\Omega)$

## 7 Conclusion

An algorithm to compute an inner approximation to the so-called one-step set for general constrained nonlinear systems is presented. The proposed algorithm is based on the interval arithmetic and provides a reliable inner approximation to the one-step set with a desired accuracy; this can be adapted to the system to be considered and it can be easily extended to the robust case. A second algorithm to obtain inner polytopes to the one-step set is presented. This representation provides simpler regions at expense of conservativeness of the approximation.

## References

Bemporad, A. (1998), ‘A predictive controller with artificial lyapunov function for linear systems with input/state

constraints’, *Automatica* **34**, 1255–1260.

Bertsekas, D. & Rhodes, I. (1971), ‘On the minmax reachability of target set and target tubes’, *Automatica* **7**, 233–247.

Blanchini, F. (1994), ‘Ultimate boundedness control for discrete-time uncertain systems via set-induced Lyapunov functions’, *IEEE Transactions on Automatic Control* **39**, 428–433.

Blanchini, F. (1999), ‘Set invariance in control’, *Automatica* **35**, 1747–1767.

Boyd, S., Ghaoui, L. E., Feron, E. & Balakrishnan, V. (1994), *Linear Matrix Inequalities in systems and control theory*, SIAM.

Cannon, M., Deshmukh, V. & Kouvaritakis, B. (2003), ‘Nonlinear model predictive control with polytopic invariant sets’, *Automatica* **39**, 1487–1494.

Cannon, M., Kouvaritakis, B. & Deshmukh, V. (2003), ‘Enlargement of polytopic terminal region in NMPC by interpolation and partial invariance’, in ‘Proceedings of the ACC’.

Chen, H. & Allgöwer, F. (1998), ‘A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability’, *Automatica* **34**(10), 1205–1218.

Chen, W., Ballance, D. & O’Reilly, J. (2001), ‘Optimisation of attraction domains of nonlinear MPC via LMI methods’, in ‘Proceedings of the ACC’.

Gilbert, E. G. & Tan, K. (1991), ‘Linear systems with state and control constraints: The theory and application of maximal output admissible sets’, *IEEE Transactions on Automatic Control* **36**, 1008–1020.

Hansen, E. (1992), *Global optimization using interval analysis*, Marcel Dekker, Inc.

Jaulin, L., Kieffer, M., Didrit, O. & Walter, E. (2001), *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer-Verlag.

Johansen, T. (2004), ‘Approximate explicit receding horizon control of constrained nonlinear systems’, *Automatica* **40**, 293–300.

Kearfott, R. (1996), *Rigorous Global Search: Continuous Problems*, Kluwer, Dordrecht, Netherlands.

Kerrigan, E. (2000), *Robust Constraint Satisfaction: Invariant Sets and Predictive Control*, PhD thesis, University of Cambridge.

Knueppel, O. (1994), ‘Profil/bias - a fast interval library’, *COMPUTING* **53**(3-4), 277–387.

Limon, D., Alamo, T. & Camacho, E. F. (2003), ‘Robust MPC based on a contractive sequence of sets’, in ‘Proceedings of the CDC’.

Limon, D., Alamo, T. & Camacho, E. F. (2004), ‘Enlarging the domain of attraction of MPC controller using invariant sets’, *Automatica*. Accepted for publication.

Mayne, D. (2001), ‘Control of constrained dynamic systems’, *European Journal of Control* (7), 87–99.

Mayne, D. Q. & Schröder, W. R. (1997), ‘Robust time-optimal control of constrained linear systems’, *Automatica* **33**(12), 2103–2118.

Moore, R. (1966), *Interval Analysis*, Prentice-Hall, Englewood Cliffs, NJ.

Rump, S. (1999), *Developments in Reliable Computing*, Kluwer Academic Publishers, chapter INTLAB - INTeRVAL Laboratory, pp. 77–104.