Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Computation and bounding of robust invariant sets for uncertain systems

H. BENLAOUKLI * S. OLARU * P. BOUCHER *

* SUPELEC, Automatic Control Department, 3 rue Joliot Curie, 91192 France (e-mail: {hichem.benlaoukli;sorin.olaru;patrick.boucher}@supelec.fr)

Abstract: This paper deals with the computational issues encountered in the construction of invariant sets of uncertainty LTI systems, the presented results being useful in the more general framework of piecewise linear systems affected by parametric uncertainty. The main contribution is the efficient computation of upper and lower bounds of the maximal positively invariant (MPI) set. These turn to be meaningful approximations when iterative construction procedures are employed, especially if no finite-time algorithms exists to construct the exact MPI set. In order to decrease the computational complexity, the interval search procedures are used to avoid the treatment of the regions which do not meet the neighboring properties.

Keywords: Model predictive control, Invariant sets, Dynamic uncertainties, Interval analysis.

1. INTRODUCTION

The set theoretic methods provide versatile tools for the engineering problems involving constraints, bounded-disturbances or construction of domains of attraction for stabilizing compensators. In this framework, the invariance is an important concept in control theory and practice, mainly due to the fact that it can be used as a versatile stability ingredient (Blanchini [1999]). Basically, the positive invariance is understood in the sense that if the system state enters in some subspace it will remain within at all future instants.

Model Predictive Control (MPC) has imposed itself as a flexible optimization based technique with constraints handling capabilities due to its time-domain formulation (Maciejowski [2002]). In the same time, the optimization fundament imposes the feasibility as a crucial demand as long as it represents the main ingredient for the stability of the entire closed loop (Mayne et al. [2000]). In this context, the positive invariance is an important ingredient assuring that the feasibility at the initial time will imply the feasibility at all future instants. Subsequently, the use of terminal constraints and the positive invariance arguments represent the current methodologies for obtaining stability guarantees.

The advances on the explicit solution for multiparametric programming made the analysis of the MPC laws easier (Bemporad et al. [2002]). Even if the terminal conditions are not imposed at the design stage, one can obtain the region of the state space where the closed loop system will perform adequately for any receding horizon control by post-processing. Different other receding horizon control schemes need to construct the maximal positively invariant set or approximate it accordingly (Gondhalekar and Imura [2007]).

Knowing that for constrained linear systems, the predictive control problems lead to piecewise affine control laws, the problem of invariant set construction is equivalent to the analysis of the stability (invariance) of piecewise affine systems. In this framework, the difficulties are originated by the fact that iterative algorithm do not offer guarantees for the decidability. This is due to the fact that the sets are not finitely parameterized (see related discussions in Gilbert and Tan [1991], Rakovic et al. [2004], Vidal et al. [2000], Alamo et al. [2005]).

The present paper propose the parallel use of two procedure which offer upper and lower approximations converging towards the maximal positively invariant (MPI) set. By using such an embedding technique, an invariant set can be obtained within a given precision. The main contribution is the use of the interval search routines (deBerg et al. [2000]) for decreasing the computational load. Without a particular care, the construction of upper and lower approximation becomes impractical due to the fact that the exploration of the possible transitions between the local linear dynamics has an exponential complexity.

In the following, section 2 recalls some basic facts about predictive control design and the explicit formulation of the control laws. Sections 3 and 4 describe in detail the construction of the maximal invariant set approximation. In section 5 the use of interval analysis for the computational amelioration is discussed while section 6 presents numerical examples illustrating the procedure. Finally the conclusions are drawn in section 7.

Notations: It is considered in the following that the polyhedron objects are bounded and can be expressed in a dual constraints-generators representation. This means that two transformations are available: $P = poly(V)$ passing from a set of vertices V to a halfspace intersection P and $V = vertice(P)$ passing from the halfspace representation P to the associated set of vertices V. Each region is defined as an intersection of a finite number of halfspaces $P_i = \{x | H_{P_i} x \leqslant K_{P_i}\}$.

The construction procedures will intensively use geometrical operations as: the union and intersection of two polyhedrons are defined respectively as $P_1 \cap P_2 = \{x | x \in P_1 \textit{ and } x \in P_2\}$ and $P_1 \cup P_2 = \{x | x \in P_1 \textit{ or } x \in P_2\}$. Polyhedron differences $P_1 \backslash P_2 = \{x | x \in P_1 \textit{ and } x \notin P_2\}$. Several software packages offer the capabilities to handle these operations in a trusty and efficient manner.

2. MODEL PREDICTIVE CONTROL. EXPLICIT FORMULATION

Let a discrete time LTI system defined by :

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) \end{cases} \tag{1}$$

In Model Predictive Control (MPC), a constrained optimal control problem over a finite receding horizon must be solved at each sample time. Generally the cost function to be minimized has the form:

$$\begin{aligned} V(k) = \; & \textstyle\sum_{i=1}^{H_p} \|y(k+i) - r(k+i)\|^2_{Q_i} \\ & + \textstyle\sum_{i=0}^{H_u-1} \|\Delta u(k+i)\|^2_{R_i} \end{aligned} \tag{2}$$

such that H_p is the prediction horizon, H_u is the control horizon and for $i \geqslant H_u$ a pre-defined stabilizing control law is used in order to construct the predictions. $Q_i = Q_i^T$ and $R_i = R_i^T$ are the weighting factor on error and control variation. $r(k+i)$ is the reference trajectory (Maciejowski [2002]). The methodology has real-time limitations for fast dynamical system and the construction of explicit solution (Bemporad et al. [2002]) may overcome the on-line computational burden. This is equivalent to the resolution of multi-parametric quadratic programs (mp-QP) (ndel et al. [2003]) where the parameters are the components of the state vector and the future references.

The feedback control law obtained for linear systems (1) with linear constraints is a piecewise affine function of the current state and the future references (Mare and Dona [2005]):

$$u_k = F_i^{FB} x_k + F_i^{FF} \begin{bmatrix} r(k+1) \\ \vdots \\ r(k+H_p) \end{bmatrix} + G_i \tag{3}$$

defined over a set of polytopic region $P_{set} = \bigcup_{i=1}^{N} P_i$ where the intersections $P_i \cap P_j$ are not full dimensional $\forall i \neq j$. F^{FB} represents the feedback gain while the F^{FF} stands for the feedforward gain which multiplies the future references considered as data for the prediction horizon. The set P_{set} represents a partition of the feasible state space and (1) becomes:

$$x_{k+1} = (A + BF_i^{FB}) x_k + BF_i^{FF} \begin{bmatrix} r(k+1) \\ \vdots \\ r(k+H_p) \end{bmatrix} + BG_i \tag{4}$$

When the reference is identically zero, the general solution (3) is projected into the solution of the regulation problem:

$$u_k = F_i x_k + G_i \tag{5}$$

and the closed loop dynamics will be given by:

$$x_{k+1} = (A + BF_i) x_k + BG_i \tag{6}$$

The stability of MPC scheme (6) is a well understood topic (Mayne et al. [2000]), guarantees can be obtained through the use of terminal constraints that assure the positive invariance of the feasible domain with respect to the closed loop dynamics. An alternative method is to use receding horizon optimization without terminal constraints, build the explicit solution and *a posteriori* determine the invariant set for the piecewise linear system (6). By using this former technique one has to obtain the invariant sets in an efficient way and in the case when the finite-time determination is not possible, to obtain upper and lower bounds within a given precision. In the next section two algorithms are proposed in this direction.

The results can be extended to the reference tracking case where the construction of invariant sets provides meaningful information about the stability for arbitrary reference. It should be mentioned that the existing stability results on this problems are restricted to specific class of signals (Limon et al. [2005]), or make use of auxiliary concepts as *reference governors* to assure the global stability (Olaru and Dumur [2005]).

3. INVARIANT SET COMPUTATION

The construction of positive invariant sets for piecewise affine systems as the one in (6) is known to be a difficult problem (see Rakovic et al. [2004]), especially due to the finite determination problems.

3.1 Outer approximation of invariant set

Let a piecewise affine autonomous dynamic system defined by (6). The following algorithm computes an upper bound $\Phi \supseteq \Phi^{MPI}$ where Φ^{MPI} is the maximal positively invariant set (MPI) of the piecewise affine system (6). In the following it is supposed that the partition P_{set} and the associated control laws are obtained using an explicit formulation of an MPC problem (see for example Kvasnica et al. [2004]) or that (6) is the model of a plant to be analyzed.

Algorithm 1. Contractive set construction

Input arguments : the matrices A and B, the polytopic regions $P_{set} = \bigcup_{i=1}^{N} \{P_i\}$, and the control laws (5) defined by F_i and G_i.
Output argument : Φ.

```
while ("precision condition" not true)
    N = cardinal(P_set)
    i = 1
    while (i ≤ N)
        [H_1, K_1] = constraint(P_i)
        po = ∅
        for j = 1, N
            [H_2, K_2] = constraint(P_j)
            h = [H_1, H_2(A + BF_i)]
            k = [K_1, K_2 − H_2 B G_i)]
            po = po ⋃ polytope(h, k)
        end
        m = cardinal(po)
        P_set = {P_1, ..., P_{i−1}, po, P_{i+1}, ..., P_N}
        i = i + m
```

(16) $N = cardinal(P_{set})$
(17) **end**
(18) **end**
(19) $\Phi = P_{set}$

The algorithm eliminates at each iteration the subset of each P_i, $i = 1, ..., N$ that leaves P_{set} in one step when applying the dynamic 6.

So for every polytope $P_i \in P_{set}$ (steps $4 \sim 17$) we calculate its subset such that its image through the affine mapping equation (6) remains within P_{set} (steps $7 \sim 12$). At each iteration (steps $1 \sim 18$), P_{set} is updated to approach the invariant set Φ^{MPI}.

Remark 1. If P_{set} do not change when it is updated (step 14), then the construction stops and the result is the exact invariant set, otherwise see the discussion in subsection 3.3.

Remark 2. When $cardinal(P_{set})$ increases, the algorithm becomes computationally expensive. Section 5 presents an amelioration based on interval analysis.

3.2 Inner approximation of invariant set

An expansive computation algorithm will proceed in reverse way comparing to algorithm 1. A lower bound for the invariant set $\Psi \subseteq \Psi^{MPI} = \Phi^{MPI} \subseteq \Phi$ will be computed starting from an initial positive invariant set (for example the maximal output admissible region - Gilbert and Tan [1991] - for the region containing the origin). Alternatively, by choosing P_1, the region in P_{set} that contain the origin and by applying algorithm 1 one gets the initial invariant set Ψ_1. The result is stored in Ψ.

Further, the idea is to compute all regions of P_{set} that transit in one step to Ψ. Iteratively Ψ is updated to contain all the sets reaching the invariant set in one step. It can be shown that Ψ expands and thus a lower bound for the maximal positively invariant set is obtained.

Algorithm 2. Expansive set construction

Input arguments : the matrices A and B, the polytopic regions $P_{set} = \bigcup_{i=1}^{N} \{P_i\}$ and the control laws (5) defined by F_i and G_i.

Output argument : Invariant set $\Psi \subset \Psi^{MPI}$.

(1) Let $P_1 \in P_{set}$ be the polytope that contains origin
(2) Use Algorithm 1 to compute the invariant set of P_1. Let the result be Ψ_1
(3) $StepSet = \Psi_1$
(4) **while**("precision condition" not true)
(5) $N = cardinal(P_{set})$
(6) $Interm = \emptyset$
(7) $L = cardinal(StepSet)$
(8) **for** $i = 1, N$
(9) $[H_1, K_1] = constraint(P_i)$
(10) $po = \emptyset$
(11) **for** $j = 1, L$
(12) $[H_2, K_2] = constraint(StepSet_j)$
(13) $h = [H_1, H_2(A + BF_i)]$
(14) $k = [K_1, K_2 - H_2 BG_i)]$
(15) $po = po \bigcup polytope(h, k)$
(16) **end**
(17) $Interm = Interm \bigcup po$
(18) **end**
(19) $\Psi = \Psi \bigcup Interm$
(20) $StepSet = Interm$
(21) **end**

Remark 3. In step (2), the result of algorithm 1 must be a non-empty set (true if the linear dynamic associated to this region is strictly stable).

Remark 4. Note that the iterative procedure described in both algorithms implies intermediate compact sets. This means that instead of treating all the regions of the partition, only the neighbors of the current set Ψ or the regions on the frontier of Φ has to be processed. Even if the number of iterations is increased, the computational effort per iteration is drastically diminished. There is however the problem of deciding which are the neighboring regions, an efficient solution being the use of the interval search (see section 5).

Remark 5. If Ψ do not change when it is updated (step 19), then the construction stops and the result is the exact positive invariant set Ψ^{MPI}, and this is one possibility to meet the "precision condition" in the algorithm. Otherwise when it is not possible to calculate the exact MPI set (non-decidability), algorithms 1 and 2 are used together to bound the exact invariant set and thus the "precision condition" will refer to the difference between the outer and the inner approximation (the volume for example) as it is discussed in the next section.

3.3 Convergence

Using the algorithmic details of the two algorithms (expansive and contractive) presented before, the following propositions demonstrate that an expansive-contractive invariant set construction converges to the maximal positively invariant set MPI.

Let for an explicit solution (5):

$$P_{set} = \bigcup_{i=1}^{N} P_i \subseteq X$$

be a collection of compact N polytopic regions, described as $P_i = \{x \in X \mid H_{P_i} x \leqslant K_{P_i}\}$, X is the working space (the region of interest in the state space, considered bounded) and suppose that in this collection of regions, P_1 is the one that contains the origin.

Proposition 6. Using the *Expansive algorithm* at any iteration step p the invariant set construction verify:

$$\Psi_p \subseteq \Psi_{p+1} \subseteq \Psi^{MPI} \tag{7}$$

where Ψ^{MPI} is the maximal positively invariant set.

Proof: The algorithm is initialized with $\Psi_1 \subseteq P_1$, an invariant set with respect to the local dynamic:

$$x_{k+1} = (A + BF_1)x_k$$

Note that in this case there is no affine part due to the fact that the region corresponds to the unconstrained optimum for the finite time control problem (2) and thus the theory of maximal output admissible sets (Gilbert and Tan [1991]) can be used for the construction of Ψ_1. This is the meaning of the step (2) of the *Expansive algorithm*, which is providing Ψ_1 in a finite number of iterations if $(A + BF_1)$ has all the eigenvalues in the unit circle. It is also evident that $\Psi_1 \subseteq \Psi^{MPI}$ as long as it represents the invariant set for a single region of the whole partition.

We dispose of Ψ_1 a non-empty initial invariant set, and define Ψ_2 as the union of all the points in P_{set} that transit in one step to Ψ_1, so:

$$\Psi_2 = \bigcup_{i=1}^{M} \psi_i \tag{8}$$

such that:

$$\psi_i = \left\{ x \in P_i \mid \begin{array}{c} H_{P_i} x \leqslant K_{P_i} \\ H_{P_1}(A + BF_{P_i})x \leqslant K_{P_1} - H_{P_1} BG_{P_i} \end{array} \right\} \tag{9}$$

Knowing that Ψ_1 is positive invariant, from equation (8) we have the inclusion $\Psi_1 \subseteq \Psi_2$ and the positive invariance of Ψ_2. Ψ^{MPI} is the maximal positively invariant set so that $\Psi_2 \subseteq \Psi^{MPI}$ with equality if and only if $\bigcup_{i=1}^{M} \psi_i = \emptyset$.

The general case of equation (7) follows by induction using the same arguments.

Proposition 7. Using the *Contractive algorithm*, at iteration p, the invariant set construction verifies:

$$\Phi^{MPI} \subseteq \Phi_{p+1} \subseteq \Phi_p \tag{10}$$

where Ψ^{MPI} is the maximal positively invariant set.

Proof: The initialisation of the *Contractive algorithm* is done by simply considering the explicit solution $\Phi_1 = P_{set}$. Then Φ_2 is defined as Φ_1 minus all polytopic regions in $\Phi_1 = P_{set}$ that transit in one step outside it, so:

$$\Phi_2 = \Phi_1 - \bigcup_{i=1}^{N} \Omega_i \tag{11}$$

such that:

$$\Omega_i = \left\{ x \in P_i \left| \begin{array}{c} H_{P_i} x \leqslant K_{P_i}; \\ \nexists j \in \{1, \ldots, N\},\ s.t. \\ H_{P_j}(A + BF_{P_i})x \geqslant K_{P_j} - H_{P_j} BG_{P_i} \end{array} \right. \right\} \tag{12}$$

By the definition of the set subtraction and the equation (11) the inclusion $\Phi_2 \subseteq \Phi_1$ is proved. $\Phi_2 = \Phi_1 = \Phi^{MPI}$ if and only if $\bigcup_{i=1}^{N} \Omega_i = \emptyset$, case when the construction procedure stops. The general case of equation (10) follows by induction.

By noting that the expansive algorithm provides at each step an invariant set while the contractive sequence is not invariant but always contain the maximal positively invariant set, we can state the next proposition which resumes the degree of approximation.

Proposition 8. $\forall p \in \mathbb{N}$, $\exists \epsilon_p^{min} \in \mathbb{R}$ such that $\Psi_p \oplus \mathbb{B}(\epsilon) \supset \Phi_p, \forall \epsilon \geqslant \epsilon_p^{min}$. Furthermore, ϵ_p^{min} is monotonically decreasing for $p \to \infty$. [1]

[1] $\oplus$ stands for the addition in Minkowski sens and $\mathbb{B}(\epsilon)$ is the ball of ray ϵ centered in the origin.

Proof: The set Ψ_1 is bounded as long as it is originated by the explicit solution of a multiparametric quadratic programme over a bounded region of the parameters space X. Then it exists ϵ_1^{min} such that $\Psi_1 \oplus \mathbb{B}(\epsilon) \supset \Phi_1, \forall \epsilon \geqslant \epsilon_1^{min}$. Then using the propositions 6 and 7, the decreasing behavior of ϵ_p^{min} is straightforward. More than that, using (7) and (10) we have $\epsilon \geqslant 0$ as long as $\Psi_p \subseteq \Psi^{MPI} = \Phi^{MPI} \subseteq \Phi_p$.

Remark 9. Both algorithms are iterating while a "precision condition" is met. This may take the form of a finite number of iterations. In this case, the ϵ_p^{min} offers information about the degree of approximation. Alternatively a threshold on ϵ^{min} can be imposed and once the algorithms go beyond this limit the expansive invariant set construction is said to have an acceptable approximation of the MPI set.

4. ROBUST INVARIANT SET COMPUTATION

Following the same idea of constructing invariant sets for linear systems controlled by piecewise affine control laws we are developing here the calculation of robust invariant set (or positive invariant approximation) for systems affected by polytopic uncertainty which can embed a large class of nonlinear systems.

Let an uncertain system defined by its nominal dynamic (developed around an operating point):

$$\begin{cases} x(k+1) = A_n x(k) + B_n u(k) \\ y(k) = Cx(k) \end{cases} \tag{13}$$

and d extreme realisations for the dynamics affected by uncertainty:

$$\begin{cases} x(k+1) = A_\Delta x(k) + B_\Delta u(k) \\ y(k) = Cx(k) \end{cases} \tag{14}$$

such that $(A_\Delta, B_\Delta) = \sum_{i=1}^{d} \alpha_i (A_i, B_i)$ and $\sum_{i=1}^{d} \alpha_i = 1$.

Before adapting the previous algorithms, we need to calculate a controller that stabilize all the dynamics of the polytopic system by using for example an LMI formulation (see Kothare et al. [1996]). This controller will be further use for imposing the terminal cost in MPC formulation for the nominal model. These precautions are necessary in order to assure that an non-empty invariant set exists.

The following two algorithms calculate or approximate the robust invariant set for this kind of systems.

4.1 Outer approximation of robust invariant set

Consider the system defined by (13) and (14). An explicit solution is obtained using the nominal system and let the result be a polytopic region P_{set} in which a controller is defined as (5). As in algorithm 1 we proceed iteratively by eliminating all subset of P_i, $i = 1, ..., N$ that goes out P_{set} for all uncertainty dynamics.

Algorithm 3. Contractive set construction

Input arguments : the matrices $\{A_1, ..., A_d\}, \{B_1, ..., B_d\}$, the polytopic regions $P_{set} = \bigcup_{i=1}^{N} \{P_i\}$ and the control laws (5) defined by F_i and G_i which is calculated using the nominal dynamic.

Output argument : $robust\Phi$.

(1) **while** ("precision condition" not true)
(2)   **for** $dyn = 1, d$
(3)   steps ($2 \sim 17$) of Algo 1 such that:
(4) $A = A_{dyn}$ and $B = B_{dyn}$
(5)   **end**
(6) **end**
(7) $robust\Phi = P_{set}$

4.2 Inner approximation of robust invariant set

Algorithm 4. Expansive set construction

Input arguments : the matrices $\{A_1, ..., A_d\}, \{B_1, ..., B_d\}$, the polytopic regions $P_{set} = \bigcup_{i=1}^{N} \{P_i\}$ and the control laws (5) defined by F_i and G_i which is calculated using the nominal dynamic.

Output argument : $robust\Psi$.

(1) Let $P_1 \in P_{set}$ be the polytope that contains origin
(2) Use Algorithm 3 to compute the robust invariant set of P_1. Let the result be $robust\Psi_1$
(3) $StepSet = robust\Psi_1$
(4) **while**("precision condition" not true)
(5) $N = cardinal(P_{set})$
(6) $Poly = \emptyset$
(7) $L = cardinal(StepSet)$
(8)   **for** $i = 1, N$
(9) $[H_1, K_1] = constraint(P_i)$
(10)   **for** $dyn = 1, d$
(11)   steps ($10 \sim 16$) of Algo 2 such that:
(12) $A = A_{dyn}$ and $B = B_{dyn}$
(13) $Poly_{dyn} = [Poly_{dyn}, po]$
(14)   **end**
(15)   **end**
(16) $robust\Psi = robust\Psi \bigcup (\bigcap_{ind=1,d} Poly_{ind})$
(17) $StepSet = robust\Psi$
(18) **end**

The difference between algorithm 4 and algorithm 2 resides in the treatment at each iteration of all the the possible system dynamics. This results in the intersection of the invariant sets (step 16) in algorithm 4 comparing with the step 19 of algorithm 2 where only the nominal model was considered.

5. COMPUTATION AMELIORATION BY INTERVAL SEARCH

Interval tree search (see deBerg et al. [2000] for basic definitions and procedures) is an algorithm which allows to efficiently identify all the intervals, in a predefined collection, that overlap a given point or interval. To do so, two important aspects has to be treated: the data structure called *interval tree* and secondly the algorithm to query it.

In the framework of the present paper, we are interested on the regions defined as unions of polytopic sets:

$$P_{set} = \bigcup_{i=1}^{N} P_i$$

Each set P_i is defined by its v_i vertices;

$$P_i = \{V_l \in \mathbb{R}^n / l = 1, ..., v_i; v_i \geq n\}$$

The projection of polytope's vertices on each dimension $j \in \mathbb{R}^n$ (see Figure 1) is thus given by

$$Proj_j(P_i) = \{V_l(j)/l = 1, ..., v_i\}$$

and allows the construction of the data structure for each dimension j:

$$I_j(P_{set}) = \begin{bmatrix} [\min(Proj_j(P_1)), \max(Proj_j(P_1))] \\ [\min(Proj_j(P_2)), \max(Proj_j(P_2))] \\ \vdots \\ [\min(Proj_j(P_N)), \max(Proj_j(P_N))] \end{bmatrix} \quad (15)$$

which corresponds to a collection of intervals characterizing P_{set}.

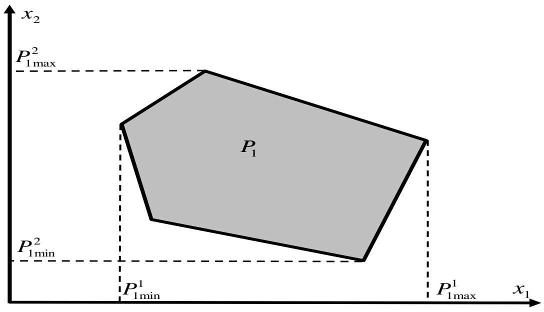

Fig. 1. Retrieving the intervals for P_1.

In the algorithm (1), the step 7 and the corresponding step in algorithm (3) are the most time-consuming parts. Indeed, this is based on the sequential scanning of all polytopes in P_{set}. In this context, the interval search can provide important ameliorations by performing the intersection between P_i and a set of *candidate polytopes* P_{can} which overlap the projection intervals on each dimension (see Fig.(2) for illustration). These candidate polytopes set is determined for a given polytope i as:

$$P_{can} = \bigcap_{j=1}^{n} Querry(I_j(P_{set}); I_j(P_{set})(i)). \quad (16)$$

where $Querry(I_j(P_{set}); I_j(P_{set})(i))$ is a function which determines, for the dimension j, all intervals (and corresponding polytopes) of $I_j(P_{set})$ that intersect:

$$[\min(Proj_j(P_i)), \max(Proj_j(P_i))]$$

Algorithms to create the interval trees and interval querying are explained in (deBerg et al. [2000]).

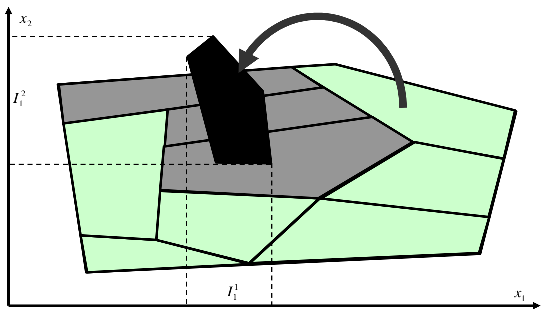

Fig. 2. In gray are represented the candidates polytopes for the intersection with the black one.

6. NUMERICAL EXAMPLE

Let an uncertain system given by its nominal dynamic:

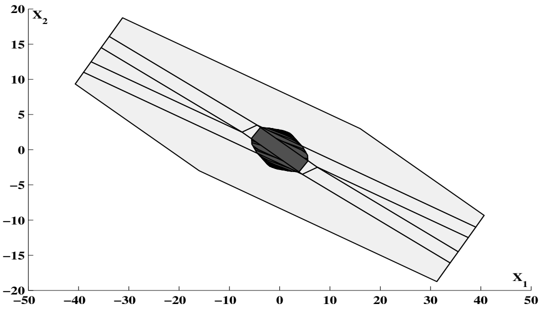

Fig. 3. Superposition of explicit solution with 9 regions and robust invariant set (black).

$$\begin{cases} x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k) \\ y(k) = [\,1 \;\, 1\,] x(k) \end{cases} \tag{17}$$

and the polytopic uncertainty:

$$\begin{aligned} (A_1, B_1) &= \left(\begin{bmatrix} 0.7 & 1 \\ 0 & 0.8 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) \\ (A_2, B_2) &= \left(\begin{bmatrix} 0.7 & 1 \\ 0 & 1.2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) \\ (A_3, B_3) &= \left(\begin{bmatrix} 1.3 & 1 \\ 0 & 0.8 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) \end{aligned} \tag{18}$$

In order to illustrate the difference between the nominal and the robust positive invariant set, a simple optimal control problem (2) with $H_p = H_u = 2$ and weights $Q = I_2$ and $R = 1$ is considered. The constraint to be satisfied are:

$$\begin{cases} -20 \leqslant y(k) \leqslant 20 \\ -1 \leqslant u(k) \leqslant 1 \end{cases} \tag{19}$$

Figure (Fig. 3) represents the superposition of the explicit solution and the invariant set of nominal system (grey color) and uncertain system (black color). The explicit solution has 3 regions and the algorithms are providing upper and lower bounds converging to the exact invariant and robust invariant set in 5 iterations due to the fact that all the possible combinations of dynamics are stable.

7. CONCLUSION

Four algorithms for finding the exact or approximate invariant and robust invariant set for piecewise affine systems were presented. The system description is supposed to be obtained as an explicit solution of a predictive control problem using the nominal dynamic. The first and third algorithms starts from this explicit solution and uses a contractive strategy to converge towards the maximal invariant and robust invariant set. On the other hand, the expansive algorithms (2nd and 4th) use an initial invariant or robust invariant set for the region containing the origin and subsequently increase it to converge towards the same maximal invariant and robust invariant set. Both algorithms 1 and 2 or algorithms 3 and 4 could be used to bound the exact invariant and robust invariant set in the case when it is not possible to find this set in a finite number of steps (non-decidability). The condition to stop the algorithms iteration may be related to the difference between the volumes of the upper and lower bounding sets. It is shown that the use of the interval search algorithm can ameliorate the computation time.

REFERENCES

T. Alamo, A. Cepeda, D. Limon, J. M. Bravo, Mirko Fiacchini, and E. F. Camacho. On the computation of robust control invariant sets for piecewise affine systems. In *International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control*, August 2005.

A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

F. Blanchini. Set invariance in control. *Automatica*, 35 (11):1747–1767, 1999.

M. deBerg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry, Algorithms and Applications*. Springer, 2000.

E.G. Gilbert and K.T. Tan. Linear systems with state and control constraints, the theory and application of maximal output admissible sets. *IEEE Transaction on Automatic Control*, 36:1008–1020, 1991.

R. Gondhalekar and J. Imura. Strong feasibility in input-move-blocking model predictive control. In *7th IFAC Symposium on Nonlinear Control Systems*, South Africa, 2007.

M. V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.

M. Kvasnica, P. Grieder, and M. Baotić. Multi-parametric toolbox (mpt), 2004. URL `http://control.ee.ethz.ch/ mpt/`.

D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho. Mpc for tracking of piece-wise constant references for constrained linear systems. In *In Proceeding 16th IFAC World Congress*, Prague, 2005.

J. M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.

J. Mare and J. De Dona. Analytical solution of constrained reference tracking problems by dynamic programming. In *In Proceedings of the IEEE Conference on Decision and Control and European Control Conference*, 2005.

D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

P. Tøndel, T. A. Johansen, and A. Bemporad. An algorithm for mpqp and explicit mpc solutions. *Automatica*, 39(3):489–497, 2003.

S. Olaru and D. Dumur. Compact explicit mpc with guarantee of feasibility for tracking. In *In Proceedings of the IEEE Conference on Decision and Control and European Control Conference*, pages 969–974, 2005.

S. V. Rakovic, P. Grieder, M. Kvasnica, D. Q. Mayne, and M. Morari. Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances. In *In Proceeding 43th IEEE Conference on Decision and Control*, December 2004.

R. Vidal, S. Schaffert, J. Lygeros, and S. Sastry. *Controlled Invariance of Discrete Time Systems*. HSCC LNCS 1790. Springer, 2000.