2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Computation of Polyhedral Positive Invariant Sets via Linear Matrix Inequalities

Daniel Rubin[1], Hoai-Nam Nguyen[2] and Per-Olof Gutman[3]

Abstract— **This paper presents a linear matrix inequality based algorithm for computation of polyhedral positive invariant sets for linear discrete-time systems subject to bounded state and input constraints. While the main algorithm is suitable for computation of polyhedral invariant sets of a general shape, a second algorithm specialized for symmetric sets is also presented. The results are extended to include polytopic uncertainty. Verification and demonstration of the proposed scheme is done through numerical examples.**

I. INTRODUCTION

Invariant sets are used in dynamical systems analysis. In particular, positive invariant sets enable control engineers to determine the behavior of closed loop systems. These sets are used in various problems such as control synthesis, stability and robustness analysis, worst case estimation, and domain of attraction computations [1]. Since only *positive* invariant sets will be discussed in this work, the word "positive" will henceforth be sometimes omitted.

Convex invariant sets are typically represented by ellipsoids or polytopes, i.e. bounded polyhedra. While the former are simple to represent and compute, the latter are more flexible. Moreover, for discrete-time linear systems under state and control constraints, the reachable sets and domain of attraction are accurately described as invariant polytopes. Polyhedral sets are also well suited for optimization, and used in standard problems such as Linear Programming (LP) and Quadratic Programming (QP). In Model Predictive Control (MPC), stability is guaranteed by using a polyhedral invariant set as a terminal set [2]. Invariant sets take an even greater part in tube-based robust MPC [3].

A new method by Nguyen and Gutman [4], [5], named *Interpolating Control*, explicitly uses invariant polyhedral sets admissible for local and global controllers, which are used to construct the control law by interpolation.

If the dynamical system is subject to parametric uncertainty, a robustly positive invariant (RPI) set is sought. Computation of RPI sets can be very demanding for systems of order greater than two. Moreover, computation of invariant sets can be numerically challenging even for certain systems of high order. The classical method by Gilbert and Tan [6], also applicable to uncertain systems, can theoretically compute the maximal admissible RPI set for any given stable linear system with constraints containing the origin. However, in practice, for high order and uncertain systems, the computation may take many hours and may require more memory than available. The resulting invariant set might be very complex, i.e. with many half-planes and vertices, thus impractical to use in on-line computations.

Considerable effort has been made in developing efficient algorithms to compute polyhedral invariant sets. In [7], Gilbert and Tan's method is improved to become more computationally effective by avoiding redundant operations. An improvement of a similar nature is also presented in [4, p. 24]. In [8] contractive ellipsoidal sets were used to obtain invariant polyhedral sets. However in many cases the contraction factor is close to unity, resulting in very complex polytopes. In [9] the Linear Matrix Inequality (LMI) optimization method is used to compute invariant polytopes without iterations. This method is not applicable to autonomous systems and only provides hypercubical polytopes. More recent works include [10], [11]. In [10] a single LP is used to compute an RPI set that is the minimal in respect to a set of inequalities with predefined normal vectors. The iterative procedure in [11] can be used to enlarge a given invariant set by adding new vertices.

This work presents a novel LMI based method for polyhedral invariant set computation. Two algorithms are included: the first algorithm, given in Section III, for general set computation, and the second, given in Section IV, for symmetrical sets. An extension for uncertain systems is given in Section V, and illustrative numerical examples in Section VII.

II. PRELIMINARIES

The problem under consideration is to find an invariant set for the autonomous system,

$$x(k+1) = Ax(k), \tag{1}$$

subject to the constraints

$$x \in \mathcal{X} = \{x \in \mathbb{R}^n : F_x x \preceq \mathbf{1}\}. \tag{2}$$

Here $\mathcal{X}$ is a polytope given in normalized half-space representation, and $\mathbf{1}$ is a column vector of ones; the $\preceq$ operator denotes element-wise inequalities.

Definition 1: A set Ω is said to be *positive invariant* w.r.t. system (1) if for each $x(k) \in \Omega$ it holds that $Ax(k) \in \Omega$.

For the uncertain case, consider the system

$$x(k+1) = A(k)x(k). \tag{3}$$

[1]Daniel Rubin is with the Autonomous Systems Program, Technion – Israel Institute of Technology, Haifa, Israel rubdaniel@campus.technion.ac.il, Supported by General Motors Company Grant 2022812

[2]Hoai-Nam Nguyen is with the Control, Signal and System Department, IFP Energies Nouvelles, Rond-point de l'échangeur de Solaize BP3, 69360 Solaize, France hoai-nam.nguyen@ifp.fr

[3]Per-Olof Gutman is with the Faculty of Civil and Environmental Engineering, Technion – Israel Institute of Technology, Haifa, Israel peo@technion.ac.il

where $A(k) \in \mathbb{R}^{n\times n}$ satisfy

$$A(k) = \sum_{i=1}^{s} \alpha_i(k) A_i, \tag{4a}$$

$$\mathbf{1}^\top \alpha = 1,\ \alpha_i \geq 0,\ \forall i = 1,\dots,s. \tag{4b}$$

Here $\alpha_i(k)$ may be an unknown constant or time varying, corresponding to (3) being an uncertain as well as a linear parameter varying (LPV) system.

Definition 2: A set Ω is said to be *robustly positive invariant* w.r.t. system (3) if for each $x(k) \in \Omega$ it holds that $A(k)x(k) \in \Omega$.

Definition 3: A set $\Omega \subseteq \mathcal{X}$, that is invariant w.r.t. (1) or robustly invariant w.r.t. (3) is said to be *constraint-admissible*.

The interested reader is referred to e.g. [1] for more details on the above standard definitions.

III. Main Algorithm

Let $V_0 \in \mathbb{R}^{n\times m}$ be a matrix of m known points in $\mathbb{R}^n$. We seek an invariant set $\Omega = \mathrm{conv}(V)$, where $V \in \mathbb{R}^{n\times m}$ is given as

$$V = V_0 \Lambda^{-1}, \tag{5}$$

with

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix}, \quad \lambda_i > 0\ \forall i = 1,\dots m. \tag{6}$$

Therefore, Ω is composed of at most m vertices lying in directions defined by the columns of V_0, that is given a priori and can be written as

$$V_0 = \begin{bmatrix} v_1 & v_2 & \cdots & v_m \end{bmatrix}. \tag{7}$$

This idea of expanding the given set of points to the largest possible invariant set by multipliying each point v_i by a scalar λ_i^{-1}, is illustrated in Fig.1.

It is well known [12] that the set Ω is invariant if and only if there exists a nonnegative square matrix[1] P such that

$$AV = VP, \tag{8}$$

and

$$P^\top \mathbf{1} \preceq \mathbf{1}. \tag{9}$$

Hence, the successor states of the vertices in V under transformation (1) can be written as convex combinations of the vertices in V, and that the sum of the elements of each column in P is ≤ 1.

Since P is nonnegative and (9) holds, P can be described as a state matrix of a *stable or marginally stable* and *positive* (autonomous) system [1, pp. 143–144]. Hence, according to [13, p. 41] there exists a positive diagonal matrix H such that

$$PHP^\top - H \leq 0. \tag{10}$$

[1]A matrix is said to be non-negative [positive] if all of its elements are non-negative [positive].

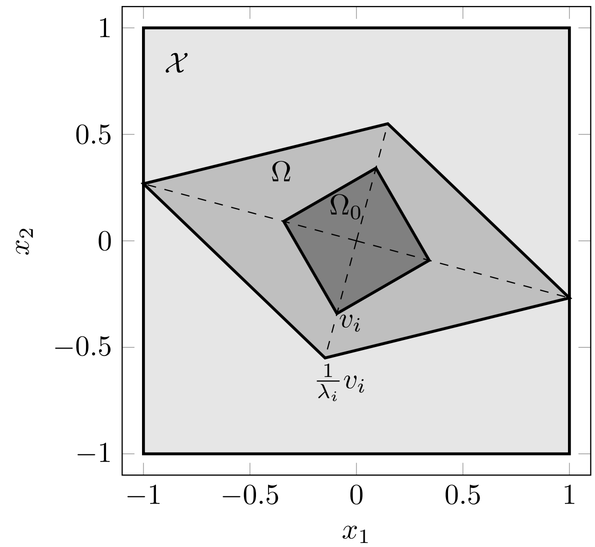

Fig. 1. 2nd order illustration of the main algorithm.

Multiplying (10) by Λ^{-1} from both the right hand side and the left hand side,

$$\Lambda^{-1}PHP^\top\Lambda^{-1} - \Lambda^{-1}H\Lambda^{-1} \leq 0. \tag{11}$$

Substituting $H = \Lambda Q \Lambda$

$$\Lambda^{-1}P\Lambda Q\Lambda P^\top\Lambda^{-1} - \Lambda^{-1}\Lambda Q\Lambda\Lambda^{-1} \leq 0, \tag{12}$$

and $Q > 0$ and diagonal. Denoting $P_0 = \Lambda^{-1}P\Lambda$ we obtain

$$P_0 Q P_0^\top - Q \leq 0. \tag{13}$$

Or,

$$P_0 Q Q^{-1} Q P_0^\top - Q \leq 0. \tag{14}$$

Eq. (14) can be rewritten using the Schur complement [14, p. 28] as

$$\begin{bmatrix} Q & P_0 Q \\ Q P_0^\top & Q \end{bmatrix} \geq 0, \quad Q > 0. \tag{15}$$

Substituting (5) into (8) we have

$$AV_0\Lambda^{-1} = V_0\Lambda^{-1}P. \tag{16}$$

Multiplying both sides by Λ from the right hand side,

$$AV_0 = V_0\Lambda^{-1}P\Lambda = V_0 P_0. \tag{17}$$

Multiplying (17) by Q from the right hand side

$$AV_0 Q = V_0 P_0 Q \tag{18}$$

Denoting $R = P_0 Q$ we obtain an LMI affine in both Q and R:

$$AV_0 Q - V_0 R = 0 \tag{19a}$$

$$\begin{bmatrix} Q & R \\ R^\top & Q \end{bmatrix} \geq 0 \tag{19b}$$

$$Q > 0 \tag{19c}$$

$$R_{ij} \geq 0 \quad \forall i, j = 1,\dots,m \tag{19d}$$

Condition (19d) comes from the fact that R is a product a non-negative matrix P and a positive diagonal matrix Q.

To have the set Ω as large as possible, i.e., a set that includes all feasible solutions of (8), it is required to minimize the elements of $P^\top \mathbf{1}$. An equivalent optimization problem is to minimize $\mathbf{1}^\top P^\top \mathbf{1}$. For a given Λ, an optimal P translates to an optimal P_0, thus one can instead minimize $\mathbf{1}^\top {P_0}^\top \mathbf{1}$. Substituting $P_0^\top = Q^{-1} R^\top$, the problem becomes,

$$\min_{Q,R} \{\mathbf{1}^\top Q^{-1} R^\top \mathbf{1}\} \tag{20}$$

subject to (19). The product $\mathbf{1}^\top Q^{-1}$ is minimal if and only if the trace$\{Q\}$ is maximal, whereas $R^\top \mathbf{1}$ is minimal if and only if the $\sum_{i=1}^m \sum_{j=1}^m R_{ij}$ is minimal. Hence, optimal Q and R are obtained by solving the semi-definite program (SDP)

$$\min_{Q,R} \left\{ -\text{trace}\{Q\} + \sum_{i=1}^m \sum_{j=1}^m R_{ij} \right\} \tag{21}$$

subject to (19).

The optimal $P_0 = RQ^{-1}$ with R and Q obtained from above optimization is used to calculate Λ as follows. Substituting $P = \Lambda P_0 \Lambda^{-1}$ into (9) we obtain

$$\Lambda^{-1} P_0^\top \Lambda \mathbf{1} \preceq \mathbf{1}. \tag{22}$$

Denote $\lambda = \Lambda \mathbf{1} = \begin{bmatrix} \lambda_1 & \cdots & \lambda_m \end{bmatrix}^\top$,

$$P_0^\top \lambda \preceq \lambda. \tag{23}$$

All points in V must be bounded by $\mathcal{X}$. Hence,

$$F_x \lambda_i^{-1} v_i \preceq \mathbf{1}, \quad \forall i = 1, \ldots, m, \tag{24}$$

with v_i the i-th column of V_0. The lower bound for every λ_i can therefore be computed by the following LP

$$\underline{\lambda}_i = \min \lambda_i \quad \text{s.t } \{-\lambda_i \mathbf{1} \preceq -F_x v_i\}. \tag{25}$$

From (23) and (25), λ can be computed by the following LP problem

$$\min_{\lambda} \sum_{i=1}^m \lambda_i \quad \text{s.t.} \quad \begin{cases} (P_0^\top - I)\lambda \preceq 0 \\ -\lambda \preceq -\underline{\lambda} \end{cases} \tag{26}$$

With the thus found Λ, (5) gives V. A summary of the method is given in Algorithm 1.

Algorithm 1 Invariant set computation

Input: Matrices A, F_x, and V_0
Output: Matrix V, for which $\Omega = \text{conv}(V)$ is invariant
$Q, R \leftarrow$ arguments of SDP (21), (19)
$P_0 \leftarrow RQ^{-1}$
for $i \leftarrow 1$ to m **do**
  $\underline{\lambda}_i \leftarrow$ solution of LP (25)
end for
$\lambda \leftarrow$ solution of LP (26)
$\Lambda \leftarrow \text{diag}(\lambda)$
$V \leftarrow V_0 \Lambda^{-1}$

IV. Zero-symmetric invariant sets

For a 0-symmetric[2] invariant set, the above algorithm can be simplified to reduce computational load. The assumed 0-symmetric $\mathcal{X}$ can be represented as

$$\mathcal{X} = \{x \in \mathbb{R}^n : \ |F_x x| \leq \mathbf{1}\}, \tag{27}$$

where $|X|$ denotes the element-wise absolute value of matrix X. We seek a 0-symmetric positive invariant set

$$\Omega = \text{conv}\left(\begin{bmatrix} V & -V \end{bmatrix}\right), \tag{28}$$

where $V \in \mathbb{R}^{n \times m_s}$ is given as $V = V_0 \Lambda^{-1}$; Λ positive diagonal matrix; V_0 is a known initial set of m_s points. The 0-symmetric Ω is invariant if and only if $\exists P \in \mathbb{R}^{m \times m}$ such that (8) holds and

$$|P^\top| \mathbf{1} \preceq \mathbf{1}. \tag{29}$$

For every real matrix P,

$$|P^\top| - P^\top \succeq 0 \ , \quad |P^\top| + P^\top \succeq 0. \tag{30}$$

Denoting $G = |P|$, we obtain

$$G^\top - P^\top \succeq 0 \ , \quad G^\top + P^\top \succeq 0. \tag{31}$$

Multiplying by Λ from the left and Λ^{-1} from the right, and substituting $P_0^\top = \Lambda P^\top \Lambda^{-1}$ and $G_0^\top = \Lambda G^\top \Lambda^{-1}$,

$$G_0^\top - P_0^\top \succeq 0 \ , \quad G_0^\top + P_0^\top \succeq 0. \tag{32}$$

By (29), G is stable. Since G is also positive, there exists a positive diagonal matrix H such that

$$GHG^\top - H \leq 0. \tag{33}$$

Hence, following (10)-(15),

$$\begin{bmatrix} Q & G_0 Q \\ Q G_0^\top & Q \end{bmatrix} \geq 0, \quad Q \geq 0, \tag{34}$$

with $Q = \Lambda^{-1} H \Lambda^{-1}$, a positive diagonal matrix.

Substituting $R_1 = G_0 Q$ and $R_2 = P_0 Q$ into (8), (32), and (34) the following LMIs are obtained:

$$AV_0 Q - V_0 R_2 = 0 \tag{35a}$$
$$R_1^\top - R_2^\top \succeq 0 \tag{35b}$$
$$R_1^\top + R_2^\top \succeq 0 \tag{35c}$$
$$\begin{bmatrix} Q & R_1 \\ R_1^\top & Q \end{bmatrix} \geq 0 \tag{35d}$$
$$Q \geq 0 \tag{35e}$$
$$R_{1ij} \geq 0 \quad \forall i, j = 1, \ldots, m \tag{35f}$$

Hence, a suitable P_0 is obtained as $P_0 = R_1 Q^{-1}$ with R and Q from the optimization (21) subject to (35). The LPs (25), (26) can be readily used with m replaced by m_s to obtain Λ, which in turn defines V. A summary of the method is given in Algorithm 2.

The LMIs (35) have three matrix variables, Q, R_1, and R_2, all in $\mathbb{R}^{m_s \times m_s}$. Since, Q is symmetric there are $m_s(2m_s+1)$ variables in total. In the general case of LMIs (19), there are

[2]A set $\mathcal{X}$ is 0-symmetric if $x \in \mathcal{X}$ implies $(-x) \in \mathcal{X}$.

$m(m+1)$ variables. A given 0-symmetric set, can be represented with half as many inequalities as a non-symmetrical set with same number of facets. Hence, $m = 2m_s$, and we have that the algorithm for 0-symmetric positive invariant set computation requires $m_s(m_s+1)/2$, i.e., number of variables was cut by half. The LPs (25), (26) also require half as many variables when substituting $m_s = m/2$.

The number of constraints is also substantially reduced: In (19) there are m^2 equality constraints, $m(m+1)$ inequality constraints, and a $(2m)\times(2m)$ matrix inequality constraint. The same constraints are replaced in (35) by $m^2/4$ equality constraints, $m(m+2)/4$ inequality constraints, and three $m\times m$ matrix inequality constraints.

Algorithm 2 0-symmetric invariant set computation

Input: Matrices A, F_x, and V_0
Output: Matrix V, for which $\Omega=\mathrm{conv}\left(\begin{bmatrix} V & -V\end{bmatrix}\right)$ is invariant

1: $Q, R_1, R_2 \leftarrow$ arguments of SDP (21), (35)
2: $P_0 \leftarrow RQ^{-1}$
3: **for** $i \leftarrow 1$ to m_s **do**
4: $\quad \underline{\lambda}_i \leftarrow$ solution of LP (25)
5: **end for**
6: $\lambda \leftarrow$ solution of LP (26)
7: $\Lambda \leftarrow \mathrm{diag}(\lambda)$
8: $V \leftarrow V_0\Lambda^{-1}$

V. Accounting for Polytopic Uncertainty

The set Ω, given in vertex representation with $V \in \mathbb{R}^{n\times m}$, is RPI w.r.t. system (3), if and only if there exists a set of s non-negative matrices P_i, such that for any $i = 1, \dots, s$,

$$A_i V = V P_i \qquad (36a)$$

$$P_i^\top \mathbf{1} \le \mathbf{1} \qquad (36b)$$

Hence, extending Algorithm 1 to include systems with polytopic uncertainties can be done by replacing (26) with

$$\min_{\lambda} \sum_{i=1}^{m} \lambda_i \quad \text{s.t.} \quad \begin{cases} (P_{01}^\top - I)\lambda \preceq 0 \\ \quad\vdots \\ (P_{0s}^\top - I)\lambda \preceq 0 \\ -\lambda \preceq -\underline{\lambda} \end{cases} \qquad (37)$$

In (37) each P_{0i} is obtained from solving optimization problem (21) subject to (19) for given A_i. Thus, (21) is to be solved s times.

Algorithm 2 is extended to account for polytopic uncertainty in a like manner where the optimization (21), (35) is solved s times.

VI. Initialization

The resulting positive invariant set from Algorithm 1 and Algorithm 2 highly depends on the initialization, i.e. the choice of initial vertices. Both the number and the direction of vectors v_i will dictate whether or not a feasible solution exists, and how close it is to the maximal positive invariant set.

The next theorem give conditions for V_0 to result in the maximal positive invariant set (MAS).

Theorem 1: Let V_∞ be the maximal constraint-admissible invariant set of (1), and let $V_0 = V_\infty M$ with M a positive diagonal matrix with entries smaller or equal to one. The algorithm will produce $V = V_\infty$.

Proof: The set V_∞ is invariant w.r.t. (1) by definition, thus $\exists P \succeq 0$ such that (8) and (9) hold. Taking $V_0 = V_\infty M$, it follows from (10)-(18) that $\exists P_0 = \Lambda^{-1}P\Lambda$ such that LMI (19) holds. Let $P_0^\star$ be the optimal solution of the SDP (19),(21), and let $V^\star$ be its pair such that (8),(9) holds. The solution is optimal in the sense that conv$(V^\star)$ includes all feasible solution of V. Hence, $P_0^\star{=}P_\infty{=}M^{-1}P_0M$, where P_∞ satisfies $AV_\infty{=}V_\infty P_\infty$.

Denote $\mu = M\mathbf{1}$. Since $V_\infty{\subseteq}\mathcal{X}$ it holds that $F_x\mu_i^{-1}v_{\infty,i} \preceq \mathbf{1}$, $\forall i$. Also, P_∞ satisfies (9). Hence by (22)-(25) it holds that μ is a feasible solution of (26). Since conv(V_∞) contains all possible V's, it holds that $\mathbf{1}\mu \le \lambda$. Hence μ is the optimal solution of (26), and $\Lambda = M$. ■

The next theorem gives a more practical method for initialization.

Theorem 2: In case the initial set Ω_0, defined as the convex hull of V_0, is invariant, Algorithm 1 will yield a feasible solution.

Proof: Since Ω_0 is invariant, every $\gamma\Omega_0 \subseteq \mathcal{X}$, with $\gamma \ge 0$, is also invariant. The matrix V, representing $\Omega = \beta\Omega_0$, thus satisfies (8), (9), with $P \succeq 0$. Taking $\Lambda = \gamma I$, it follows from (10)-(18) that $\exists P_0 = P$ such that (19) holds.

Then, $\beta\mathbf{1}$ is a solution to (26). Since $\Omega_0 \subseteq \mathcal{X}$, we have that by minimizing trace$\{\Lambda\}$ the solution of (26), yields $0 \le \gamma \le 1$. ■

Hence, by initializing with an invariant Ω_0, a feasible solution is obtained, which is in the worst case $\Omega = \Omega_0$. There exist known methods to construct an initial invariant set, e.g. [8] and [10]. Moreover, the presented algorithms can be used iteratively, where each iteration is initialized with

$$V_0(j) = [V(j-1) \quad \bar{V}_0(j)]. \qquad (38)$$

That is, $V_0(j)$ is the initial points matrix for iteration j, includes the results from the previous iteration and a new set of points $\bar{V}_0(j)$.

VII. Numerical Examples

This section presents two examples that serves to show the usage and benefits of the presented algorithms. The examples were computed using Matlab 2014a on an Intel Core i7-4800MQ, 32GB of memory, Windows computer. The SDPs were solved with aid of the CVX modeling package [15], [16], and the SDPT3 solver [17], [18]. The LPs were solved using the CDD solver [19], [20]. The results were visualized using the Multi-Parametric Toolbox (MPT) [21].

A. Example 1: Second order system

Consider the constrained system

$$\begin{cases} x(k+1) = \begin{bmatrix} 1 & 0.1 \\ 0 & 0.98 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0.98 \end{bmatrix} u(k), \\ -1 \le x_1 \le 1, \quad -1 \le x_2 \le 1, \\ -0.1 \le u \le 0.1. \end{cases} \qquad (39)$$

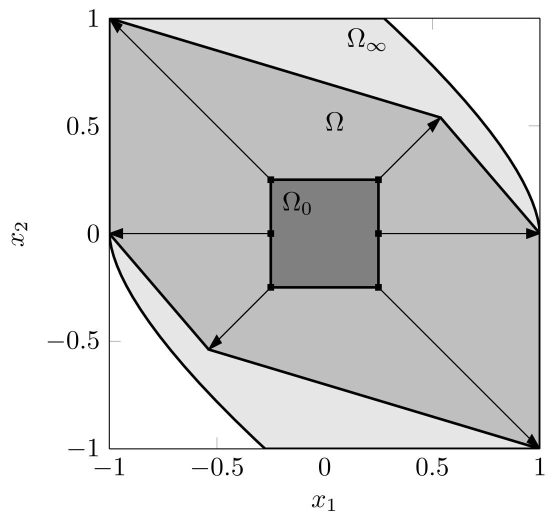

Fig. 2. Results of Example 1. Ω_∞ (light), Ω (darker), and Ω_0 (darkest). Arrows illustrates the progression of points in Ω_0 to vertices in Ω.

A stabilizing feedback controller is given as an LQR controller with $Q = I_2$ and $R = 0.001$. The MAS of the resulting closed loop (autonomous) system, denoted as Ω_∞, is obtained using Gilbert and Tan's algorithm [6] after 18 iterations, and is composed of 38 vertices.

Algorithm 2 was used with

$$V_0 = 0.25 \begin{bmatrix} -1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix},$$

resulting in

$$V = \begin{bmatrix} -0.53856 & -1 & 1 \\ -0.53856 & 1 & -1 \end{bmatrix}.$$

It is left for the interested user to find that Algorithm 1, initialized with $\tilde{V}_0 = [V_0 - V_0]$ yields the same result. Fig. 2 shows the MAS Ω_∞ together with $\Omega_0 = \text{conv}\left(\begin{bmatrix} V_0 & -V_0 \end{bmatrix}\right)$ and $\Omega = \text{conv}\left(\begin{bmatrix} V & -V \end{bmatrix}\right)$.

B. Example 2: Third order uncertain system

Consider the constrained uncertain system

$$\begin{cases} x(k+1) = \begin{bmatrix} a_{11}(k) & a_{12}(k) & a_{13}(k) \\ a_{21}(k) & a_{22}(k) & a_{23}(k) \\ 1 & 0 & 1 \end{bmatrix} x(k) \\ |86.9\,x_1 + 1205.9\,x_2 + 1.4\,x_3| \le 1, \\ |6.9\,x_1 - 11.4\,x_2 + 0.2\,x_3| \le 1, \\ |4.2\,x_3| \le 1. \end{cases} \tag{40}$$

The parameters $a_{ij}(k)$ are unknown and defines the uncertainty structure of (4) with $s = 8$ by the extreme points given in Table I. The Conversion to (4) is done straightforward by substituting the parameters in row i to obtain A_i.

Initial points are given in the 0-symmetric description by

$$V_0 = 10^{-2} \cdot \begin{bmatrix} -0.1 & -0.07 & 0.99 & 0.03 \\ -0.01 & 0 & 0.09 & 0 \\ 1 & 1 & -0.12 & 1 \end{bmatrix}.$$

TABLE I

EXTREME POINTS FOR POLYTOPIC UNCERTAINTY IN EXAMPLE 2.

point	a_{11}	a_{21}	a_{12}	a_{22}	a_{13}	a_{23}
1	0.653	-0.0133	1.42	0.978	-0.00891	-0.000294
2	0.685	-0.0124	1.37	0.981	-0.00799	-0.000262
3	0.656	-0.0135	1.37	0.981	-0.00891	-0.000296
4	0.684	-0.0124	1.38	0.981	-0.00799	-0.000262
5	0.683	-0.0123	1.42	0.979	-0.00799	-0.000262
6	0.653	-0.0134	1.42	0.979	-0.00891	-0.000296
7	0.662	-0.0122	1.71	0.978	-0.00831	-0.00026
8	0.625	-0.0133	1.8	0.979	-0.00939	-0.000293

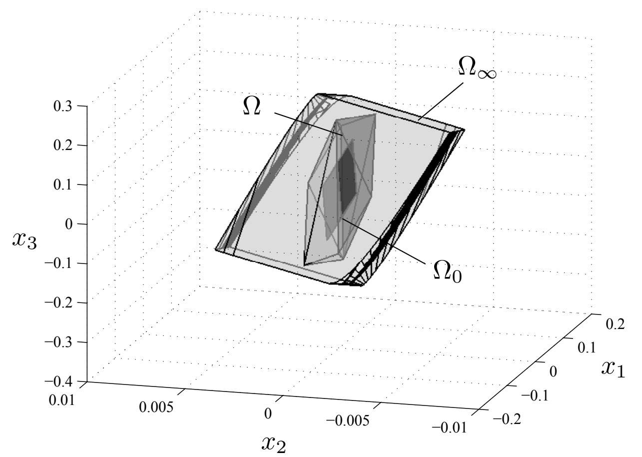

Fig. 3. Results of Example 2. Ω_∞ (light), Ω (darker), and Ω_0 (darkest).

The two presented algorithms delivered the same 0-symmetric invariant set Ω, that has eight vertices.

The standard algorithm by Gilbert and Tan took about 71 seconds to compute the MAS Ω_∞, which is composed of 296 vertices. Our approach however delivered an admissible, simple, RPI set in about 2.5 seconds by Algorithm 2. Algorithm 1 initialized with $\begin{bmatrix} V_0 & -V_0 \end{bmatrix}$ yields the same result within 5 seconds. Fig. 3 shows the MAS Ω_∞ together with the initial set $\Omega_0 = \text{conv}(V_0)$ and the optimized RPI set $\Omega = \text{conv}(V)$.

VIII. CONCLUSIONS

This paper presented a novel algorithm for computation of invariant sets for the class of discrete-time constrained systems with polytopic uncertainty. The algorithm is based on the solution of one (or multiple in the uncertain case) SDP(s) and two LPs. When uncertainty is included, the computation time increase is proportional to the number of extreme cases in the polytopic uncertainty representation.

The resulted invariant set is of a given number of vertices that can be selected as small. Not only the number but also the directions of the initial points must be taken into account. The algorithms can be used iteratively to enlarge the set obtained in each step, or in a sequence of separate runs, each with different initial points, after which the invariant set is formed as the convex hull of all of the results.

The algorithm is demonstrated in two illustrative numerical examples. Compared to the standard approach of

MAS computation, the new method is shown to be more efficient in time for high dimensional systems with or without uncertainties.

The algorithm is however still impractical for systems of large dimension and uncertainty as the computation time becomes too large. Including complex polytopes of many vertices is also not feasible due to memory demands by the SDP solvers. Current research is focused on addressing these issues as well as extending the method for systems with additive disturbances.

REFERENCES

[1] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*, ser. Systems & Control: Foundations & Applications. Boston: Birkhäuaser, 2007.

[2] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.

[3] W. Langson, I. Chryssochoos, S. V. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[4] H.-N. Nguyen, *Constrained control of uncertain, time-varying, discrete-time systems*, ser. Lecture Notes in Control and Information Sciences, M. Thoma, F. Allgöwer, and M. Morari, Eds. Springer, 2014, vol. 451.

[5] H.-N. Nguyen, P.-O. Gutman, S. Olaru, and M. Hovd, "Implicit improved vertex control for uncertain, time-varying linear discrete-time systems with state and control constraints," *Automatica*, vol. 49, no. 9, pp. 2754–2759, 2013.

[6] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and application of maximal output admissible sets," *IEEE Trans. Autom. Control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[7] B. Pluymers, J. A. Rossiter, J. A. K. Suykens, and B. De Moor, "The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty," in *Proc. 2005 American Control Conf.* Portland, OR: IEEE, 2005, pp. 804–809.

[8] A. Alessio, M. Lazar, A. Bemporad, and W. P. M. H. Heemels, "Squaring the circle: An algorithm for generating polyhedral invariant sets from ellipsoidal ones," *Automatica*, vol. 43, no. 12, pp. 2096–2103, 2007.

[9] F. Tahir, "Efficient computation of robust positively invariant sets with linear state-feedback gain as a variable of optimization," in *Proc. 2010 7th Int. Conf. Electrical Engineering Computing Science and Automatic Control*. IEEE, Sep. 2010, pp. 199–204.

[10] P. Trodden, "A one-step approach to computing a polytopic robust positively invariant set," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4100–4105, Dec. 2016.

[11] G. Bitsoris and M. Vassilaki, "Enlargement of contractive sets of discrete-time systems: The generators approach," in *Proc. 25th Mediterranean Conf. Control and Automation (MED)*. IEEE, 2017, pp. 174–179.

[12] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, pp. 1747–1767, 1999.

[13] L. Farina and S. Rinaldi, *Positive Linear Systems*. Hoboken, NJ, USA: John Wiley and Sons, Inc., 2000.

[14] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, ser. SIAM studies in applied mathematics. Society for Industrial and Applied Mathematics, 1994, vol. 15.

[15] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[16] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110.

[17] K. C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3 – a MATLAB software package for semidefinite programming," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 545–581, 1999.

[18] R. H. Tütüncü, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Mathematical Programming*, vol. 95, no. 2, pp. 189–217, Feb. 2003.

[19] K. Fukuda, "CDD," http://www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html, 2005.

[20] F. Torrisi and M. Baotic, "CDDMEX – Matlab interface for the CDD solver," http://control.ee.ethz.ch/~cohysys/cdd.php, 2005.

[21] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. European Control Conf.*, Zürich, Switzerland, July 17–19 2013, pp. 502–510.