Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 05-172

Construction of Reduced Complexity Polyhedral Invariant Sets for LPV Systems using Linear Programming [1]

B. Pluymers[2] , J.A.K. Suykens[2] and B. De Moor[2]

August 2005

Submitted for publication

[1]This report is available by anonymous ftp from *ftp.esat.kuleuven.be* in the directory *pub/sista/pluymers/reports/AUT06_reduced_complexity.pdf*

[2]K.U.Leuven, Dept. of Electrical Engineering (ESAT), Research group SCD-SISTA, Kasteelpark 10, 3001 Leuven, Belgium, Tel. 32/16/32 10 35, Fax 32/16/32 19 70, WWW: *http://www.esat.kuleuven.be/scd.* E-mail: {*bert.pluymers,johan.suykens,bart.demoor*}*@esat.kuleuven.ac.be.*

Research supported by Research Council KUL: GOA-Mefisto 666, GOA-Ambiorics, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects, G.0240.99 (multilinear algebra), G.0407.02 (support vector machines), G.0197.02 (power islands), G.0141.03 (Identification and cryptography), G.0491.03 (control for intensive care glycemia), G.0120.03 (QIT), G.0452.04 (QC), G.0499.04 (robust SVM), research communities (ICCoS, ANMMM, MLDM); AWI: Bil. Int. Collaboration Hungary/ Poland; IWT: PhD Grants, GBOU (McKnow) Belgian Federal Government: Belgian Federal Science Policy Office: IUAP V-22 (Dynamical Systems and Control: Computation, Identification and Modelling, 2002-2006), PODO-II (CP/01/40: TMS and Sustainibility); EU: FP5-Quprodis; ERNSI; Eureka 2063-IMPACT; Eureka 2419-FliTE; Contract Research/agreements: ISMC/IPCOS, Data4s, TML, Elia, LMS, IPCOS, Mastercard; Bert Pluymers is a research assistant with the I.W.T. (Flemish Institute for Scientific and Technological Research in Industry). Dr. Johan Suykens is an associate professor at the Katholieke Universiteit Leuven. Dr. Bart De Moor is a full professor at the Katholieke Universiteit Leuven, Belgium.

Abstract

This paper deals with the construction of polyhedral invariant sets for Linear Parameter-Varying (LPV) systems with polytopic uncertainty description. An efficient algorithm for exactly constructing the Maximal Admissible Set (MAS, Gilbert *et al.*, 1991) for such systems was recently proposed in (Pluymers *et al.*, 2005). However, for larger dimensional systems the number of constraints describing the MAS typically increases exponentially, which makes the construction of these sets computationally intractable. This paper first discusses the two mechanisms that can lead to the large number of constraints and then provides two methods to counter these mechanisms. The first method (called *pruning*) uses constraint tightening in order to make other constraints redundant and thereby reducing the number of branches in the tree structure of the invariant set. The second method (called *trimming*) is based on a new theorem that allows the invariant set to be computed for modified system matrices without sacrificing invariance with respect to the original system matrices. This additional freedom allows the depth of the tree structure of the invariant set to be reduced. Both methods lead to polyhedral invariant sets that are described by a reduced number of constraints compared to the MAS, but at the cost of a volume reduction. Several examples are provided to illustrate the efficiency of both methods.

Construction of Reduced Complexity Polyhedral Invariant Sets for LPV Systems using Linear Programming ⋆

B. Pluymers, J.A.K. Suykens, B. De Moor

Katholieke Universiteit Leuven, ESAT-SCD-SISTA,
Kasteelpark Arenberg 10, B-3000 Leuven (Heverlee), Belgium
E-Mail : {bert.pluymers, johan.suykens, bart.demoor}@esat.kuleuven.be

Abstract

This paper deals with the construction of polyhedral invariant sets for Linear Parameter-Varying (LPV) systems with polytopic uncertainty description. An efficient algorithm for exactly constructing the Maximal Admissible Set (MAS, Gilbert *et al.*, 1991) for such systems was recently proposed in (Pluymers *et al.*, 2005). However, for larger dimensional systems the number of constraints describing the MAS typically increases exponentially, which makes the construction of these sets computationally intractable. This paper first discusses the two mechanisms that can lead to the large number of constraints and then provides two methods to counter these mechanisms. The first method (called *pruning*) uses constraint tightening in order to make other constraints redundant and thereby reducing the number of branches in the tree structure of the invariant set. The second method (called *trimming*) is based on a new theorem that allows the invariant set to be computed for modified system matrices without sacrificing invariance with respect to the original system matrices. This additional freedom allows the depth of the tree structure of the invariant set to be reduced. Both methods lead to polyhedral invariant sets that are described by a reduced number of constraints compared to the MAS, but at the cost of a volume reduction. Several examples are provided to illustrate the efficiency of both methods.

Key words: Set Invariance, Linear Programming, Parameter-Varying Systems

1 Introduction

In recent years, set invariance has become an important tool for the analysis of dynamic systems subject to input and/or state constraints and for the synthesis of constrained control algorithms. Feasible invariant sets characterize the set of initial states for which a dynamic system is guaranteed to satisfy the imposed constraints. Set invariance is the ubiquitous tool for the synthesis and analysis of constrained controllers and in recent years specifically for Modelbased Predictive Control (MPC) [1,9,11,12,15,17]. An overview of set invariance can be found in [2].

Whether used as an analysis or a synthesis tool, it is typically favorable – in order to minimize conservatism – to construct the invariant set with maximal volume, called the Maximal Admissible Set (MAS). This concept was initially proposed in [6] for LTI systems and was shown to be well-posed, in the sense that for stable systems, such a set is uniquely defined and consists of all initial states that lead to feasible trajectories. However, the straightforward extension of these results towards LPV systems remained largely unexplored due to the apparent combinatorial explosion of the number of constraints describing the MAS. Ellipsoidal and parallellotopic invariant inner approximations of the MAS were proposed in [8] and [4,10] respectively. However, due to their restricted shape, these sets can be overly conservative. In addition, ellipsoidal invariant sets can lead to computationally complex on-line computations when used for controller synthesis, while the latter can only be constructed for a limited class of stable systems. Recent results [5] use partial invariance to obtain larger invariant sets of low complexity but are still restricted to the same class of systems.

Two different approaches can be identified for constructing reduced complexity polyhedral invariant sets. One approach consists of imposing a fixed low complexity structure on the invariant set and then scaling up the set in order to obtain a maximal volume without losing the invariance property. The forementioned results [4,5,8,10]

⋆ This paper was not presented at any IFAC meeting. Corresponding author Bert Pluymers, E-mail bert.pluymers@esat.kuleuven.be, Tel. +32-16-321709

can be classified in this category. A different diametrically opposed approach starts from a full complexity invariant set (i.e. the MAS) which is then modified in order to reduce the number of constraints. The latter approach is the one used in this paper. Recent results [14] have shown that the MAS of an LPV system can be computed exactly in a computationally efficient way, by solving a set of linear programs (LP's), if the number of linear inequality constraints describing the MAS is sufficiently small. However, this assumption tends not to hold for larger dimensional systems and therefore in this paper two modifications to this algorithm are proposed in order to reduce the number of constraints describing the resulting set. This is achieved without the need for explicitly constructing the MAS first, which results in a significant computational advantage.

This paper has two main contributions. First of all the structure of the MAS for LPV systems is discussed. It is shown that the constraints of the MAS can be classified in a tree structure, which is a generalization of the structure of the MAS for LTI systems described in [6]. Based on this tree structure, the two mechanisms that can lead to a large number of constraints – *branching* and *tree depth* – are then discussed. Finally, a modified invariance condition is then formulated that provides insights into how this tree structure can be modified without losing invariance. These insights enable the formulation of new algorithms for the construction of invariant sets for LPV systems, of which the second part of this paper is one possible embodiment.

The second contribution of this paper consists of two methods that allow to construct invariant sets with a reduced number of constraints. The first method (pruning) uses constraint tightening in order to eliminate branching in the tree structure of the invariant set. The second method (trimming) modifies the system matrices used in the construction of the invariant set in order to reduce the depth of the tree. Both methods can be combined in order to obtain an optimal trade-off between minimal complexity and maximal volume of the constructed invariant set. Tuning guidelines for the initial choice of relevant algorithm parameters are also provided.

This paper is organized as follows. In Section 2 the necessary notation and background regarding set invariance is introduced. Section 3 then gives some more insights in the structure of the MAS. These insights are then used in Section 4 to reduce the number of branches in the constraint tree by means of constraint tightening, while Section 5 shows how the depth of the constraint tree can be reduced by calculating the invariant set for modified system matrices. Section 6 then gives several examples illustrating the new methods and Section 7 finally states the conclusions and future work.

Throughout this paper MATLAB notation is used to denote rows, columns and elements of matrices, i.e. $A_{[i,:]}$ denotes the i-th row of matrix A and $A_{[i,j]}$ denotes the element at row i and column j of matrix A. The notation $[A_1;\ A_2]$ will be used to denote stacked matrices $[A_1^{\mathrm{T}}\ A_2^{\mathrm{T}}]^{\mathrm{T}}$.

2 Background

2.1 Problem Formulation

In this paper we consider autonomous LPV systems of the form

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

with $x(k) \in \mathbb{R}^{n_x}$ denoting the state of the system at discrete time k. $\Phi(k)$ represents the dynamic behaviour at time k and belongs to a given convex uncertainty polytope

$$\Omega = \left\{ \Phi \in \mathbb{R}^{n_x \times n_x} \,\middle|\, \Phi = \sum_{i=1}^{L} \lambda_i \Phi_i \ , \sum_{i=1}^{L} \lambda_i = 1, \lambda_i \geq 0 \right\}. \tag{2}$$

This system typically describes a closed loop system of an LPV system and a linear state feedback law, but the results in this paper are not restricted to this setting. The state vector is subject to linear constraints

$$x(k) \in \mathcal{X} = \{x | A_x x \leq 1_{\mathrm{v}}\}, \qquad k = 0, \ldots, \infty, \tag{3}$$

where 1_{v} denotes a column matrix consisting of only 1's and $\mathcal{X}$ is a compact set. Furthermore, we assume that the system is robustly asymptotically stable

$$\lim_{n \to \infty} \max_{x(0) \in \mathcal{X}, \Phi(k) \in \Omega, k=0,\ldots,n-1} \|x(n)\|_2 = 0. \tag{4}$$

We now introduce the concept of robust positive invariance [6].

Definition 1 (Robust Positive Invariance) *Given a system* (1)-(2) *satisfying* (4) *then* $\mathcal{S} \in \mathbb{R}^{n_x}$ *is a robust positive invariant set iff*

$$\Phi x \in \mathcal{S}, \quad \forall x \in \mathcal{S}, \forall \Phi \in \Omega. \tag{5}$$

Definition 2 (Feasibility) *An invariant set* $\mathcal{S}$ *for a system* (1)-(2) *is* feasible *with respect to constraints* (3) *if* $\mathcal{S} \subseteq \mathcal{X}$.

Remark 1 *Combining both definitions it is clear that all initial states belonging to a feasible positive invariant set have feasible corresponding trajectories. The term* positive *refers to the fact that only future trajectories are guaranteed to satisfy the constraints. In further discussions positive invariance will be referred to as invariance without explicitly using the term* positive.

We can now state two related problem formulations.

Problem 1 (P1) *Given a system* (1)-(2) *satisfying* (4) *and given state constraints* (3)*, find the feasible and robust positive invariant set* $\mathcal{S}$ *with maximal volume.*

This problem was tackled in [13] and [14]. Similar results were reported in [3]. The largest volume feasible positive invariant set, also called the Maximal Admissible Set (MAS, [6]), has been shown to be uniquely defined, polyhedral and can be shown to consist of all initial states for which all possible future trajectories are guaranteed to satisfy the imposed constraints.

Problem 2 (P2) *Given a system* (1)-(2) *satisfying* (4) *and given state constraints* (3)*, find a polyhedral feasible and robust positive invariant set* $\mathcal{S}$ *with maximal volume described by as few inequalities as possible.*

2.2 Maximal Admissible Sets for LPV systems

In this section the results from [14] are briefly summarized. The algorithm there proposed exactly constructs the MAS for LPV systems and therefore can be seen as one extremum in the spectrum of algorithms that make a trade-off between maximal volume and minimal complexity.

We first reformulate a necessary and sufficient condition for robust positive invariance, after which we can formulate an efficient algorithm for constructing the MAS for a given LPV system subject to linear inequality constraints.

Proposition 1 *A set* $\mathcal{S}$ *is a robust positive invariant set for an LPV system* (1)-(2) *iff*

$$\mathcal{S} \subseteq \mathcal{S}^{-}, \tag{6}$$

with the $^{-}$*-operator defined as*

$$\mathcal{S}^{-} \triangleq \{x | \Phi x \in \mathcal{S}, \forall \Phi \in \Omega\}. \tag{7}$$

Proof : For brevity we refer to [14] for the proof. □

Remark 2 $\mathcal{S}^{-}$ *consists of all previous states for which it is guaranteed that the current state lies within* $\mathcal{S}$ *and is therefore sometimes referred to as the pre-set of* $\mathcal{S}$*. If* $\mathcal{S}$ *is a polyhedral set defined by matrix* $A_{\mathcal{S}}$*, the pre-set can be calculated as*

$$\mathcal{S}^{-} = \{x | A_{\mathcal{S}}\Phi_1 \leq 1_{\mathrm{v}}, \ldots, A_{\mathcal{S}}\Phi_L \leq 1_{\mathrm{v}}\}. \tag{8}$$

Before stating the algorithm introduced in [14] the concept of *constraint significance* is first introduced. This concept will also be used intensively in section 4.

Definition 3 *Given a set* $\mathcal{S} \subset \mathbb{R}^n$ *and a linear constraint* $a^{\mathrm{T}}x \leq 1$ *with* $x \in \mathbb{R}^n$ *then the significance of* a *with respect to* $\mathcal{S}$ *is defined as*

$$\mathrm{sig}_{\mathcal{S}}(a) \triangleq \max_{x \in \mathcal{S}} a^{\mathrm{T}}x. \tag{9}$$

A significance $\mathrm{sig}_{\mathcal{S}}(a) > 1$ indicates that intersecting $\mathcal{S}$ with $a^{\mathrm{T}}x \leq 1$ would decrease $\mathcal{S}$ in volume. In the other case the constraint $a^{\mathrm{T}}x \leq 1$ is called *redundant* with respect to $\mathcal{S}$:

Lemma 1 *Iff* $\mathrm{sig}_{\mathcal{S}}(a) \leq 1$ *then* $a^{\mathrm{T}}x \leq 1$ *is redundant with respect to* $\mathcal{S}$*, i.e.* $\mathcal{S} \cap \{x | a^{\mathrm{T}}x \leq 1\} \equiv \mathcal{S}$*.*

Proof : If $\mathrm{sig}_{\mathcal{S}}(a) \leq 1$ all points in $\mathcal{S}$ will satisfy the constraint $a^{\mathrm{T}}x \leq 1$ which proves the lemma. □

Remark 3 *For the sake of brevity, in future sections, when* $\mathcal{S}$ *is a polyhedral invariant set and* $a^{\mathrm{T}}x \leq 1$ *is one of the defining constraints of* $\mathcal{S}$*, the constraint* $a^{\mathrm{T}}x \leq 1$ *will be called redundant with respect to* $\mathcal{S}$ *if it is redundant with respect to the set obtained when removing the constraint* $a^{\mathrm{T}}x \leq 1$ *from the definition of* $\mathcal{S}$*.*

Now an efficient algorithm for constructing the MAS for system (1)-(2) subject to constraints (3) can be formulated :

Algorithm 1 (MAS for LPV systems) *Given an LPV system* (1)-(2) *subject to constraints* (3)*, perform the following operations (throughout the algorithm the set* $\{x | A_{\mathcal{S}}x \leq 1_{\mathrm{v}}\}$ *will be denoted as* $\mathcal{S}$*) :*

Initialize $A_{\mathcal{S}} := A_x$*,* $i := 1$ *and iterate while* i *is not strictly larger than the number of rows in* $A_{\mathcal{S}}$*:*

(1) Define $a^{\mathrm{T}} = (A_{\mathcal{S}})_{[i,:]}$*.*
(2) For $j = 1, \ldots, L$*, if* $\mathrm{sig}_{\mathcal{S}}(\Phi_j^{\mathrm{T}}a) > 1$ *then set* $A_{\mathcal{S}} := [A_{\mathcal{S}};\ a^{\mathrm{T}}\Phi_j]$*.*
(3) Set $i := i + 1$*.*

Return $\mathcal{S} = \{x | A_{\mathcal{S}}x \leq 1_{\mathrm{v}}\}$*.*

This algorithm can be proven to converge if the system (1)-(2) is quadratically stable, i.e.

$$\exists P = P^{\mathrm{T}} > 0 : \quad \Phi_j^{\mathrm{T}} P \Phi_j < P,\ j = 1, \ldots, L, \tag{10}$$

and can be shown to construct the MAS for the given system and constraints. We refer the reader to [14] for a full proof. Such systems can be similarly transformed into a system defined by matrices Φ_i' such that $\|\Phi_i'\| \leq 1$, by using $\Phi_i' = V\Phi_i V^{-1}$, with $V^2 = P$. In further sections we will assume that systems are transformed in such a way before calculating the invariant set.

Remark 4 *For efficiency it is recommendable to recheck all constraints for redundance regularly during the execution of the algorithm in order to reduce the associated computational overhead.*

The proposed algorithm typically has a polynomial computation time as a function of the total number of constraints describing the MAS $\mathcal{S}$. However, as the state dimension n_x increases, the number of constraints describing $\mathcal{S}$ typically increases exponentially, as will be shown in Section 6. Therefore the computation of the MAS for higher dimensional systems (e.g. > 5) can become intractable. Therefore, in the following sections, two new algorithms are introduced that construct lower complexity invariant inner approximations of the MAS.

3 Structure of the MAS

It is clear from the recursive nature of Algorithm 1 that there is a relationship between the different inequalities describing the MAS. The matrix $A_{\mathcal{S}}$ defining the MAS $\mathcal{S}$ can be rewritten as

$$A_{\mathcal{S}} = [T_0;\ T_1;\ \dots\ T_n], \tag{11}$$

with T_0 containing those constraints from A_x that are non-redundant with respect to the constraints from $T_1, \dots, T_n$ and where, for $i = 0, \dots, n-1$, each row a^{T} of T_{i+1} can be written as $b^{\mathrm{T}}\Phi_j$ with b^{T} a row of T_i and $j \in \{1, 2, \dots, L\}$. Similarly, using set notation, the MAS $\mathcal{S}$ can be written as

$$\mathcal{S} = \bigcap_{i=0}^{n} \mathcal{T}_i, \tag{12}$$

where

$$\mathcal{T}_i = \{x | T_i x \leq 1_{\mathrm{v}}\}, \qquad i = 0, \dots, n. \tag{13}$$

This allows the use of trees to depict the structure of a MAS, where the constraints of the different sets $\mathcal{T}_i$ are located at different depths in the tree. See Figure 1 for a depiction of an invariant set for a simple numerical example.

This explicit decomposition of the set $\mathcal{S}$ allows us to rewrite the invariance condition (6) as follows :

$$\mathcal{S} \subseteq \mathcal{T}_i^-, \qquad i = 0, \dots, n. \tag{14}$$

Remark 5 *The sets $\mathcal{T}_i$ can be related to the partially invariant sets $\Omega^{(-i)}$ discussed in [5], however in this paper it is not required that $\mathcal{T}_0$ is a subset of an invariant set, whereas this is a requirement for the set $\Omega^{(0)}$.*

We will now reformulate Algorithm 1 as a first step towards formulating an algorithm for constructing reduced complexity invariant sets. In what follows we will use sets $\mathcal{S}_i$ defined as follows :

$$\mathcal{S}_i = \bigcap_{k=0}^{i} \mathcal{T}_k, \qquad i = 0, \dots, n. \tag{15}$$

Algorithm 2 (Structured MAS for LPV systems) *Given an LPV system* (1)-(2) *subject to constraints* (3)*, perform the following operations :*

(1) Set $\mathcal{T}_0 := \mathcal{X}$ and $n := 0$.
(2) Iterate until $\mathcal{S}_n \subseteq \mathcal{T}_n^-$:
  (a) Set $\mathcal{T}_{n+1} := \mathcal{T}_n^-$
  (b) Remove constraints from $\mathcal{T}_0, \dots, \mathcal{T}_{n+1}$ that are redundant with respect to $\mathcal{S}_{n+1}$
  (c) Set $n := n + 1$.

Return $\mathcal{S} = \mathcal{S}_n = \bigcap_{i=0}^{n} \mathcal{T}_i$.

Remark 6 *Note that step 2a) can be replaced by choosing $\mathcal{T}_{n+1}$ such that $\mathcal{T}_{n+1} \subset \mathcal{T}_n^-$ instead of $\mathcal{T}_{n+1} = \mathcal{T}_n^-$. It is clear that, if the algorithm still converges in this case, condition* (14) *is still satisfied and that consequently the resulting set $\mathcal{S}$ is also invariant. The next section explains how this additional flexibility can be used to reduce the number of constraints describing the set $\mathcal{S}$.*

4 Branch reduction : invariant set construction using *pruning*

In this section an algorithm is introduced that constructs an invariant set in a similar way to Algorithm 1 but uses constraint tightening in order to reduce the number of constraints describing the invariant set. The first subsection describes how constraints can be tightened without losing the invariance property. In the second subsection we then formulate an algorithm that makes use of these insights to reduce the number of inequalities describing the invariant sets. A third subsection gives rules of thumb for parameter tuning.

4.1 Constraint tightening

Based on the insights from the previous section it is now possible to formulate a strategy for constructing invariant sets with decreased complexity.

We will use Figure 1 to explain how constraint tightening can eliminate branches in the tree structure. When inspecting Figure 1 one can notice that two separate branches appear below constraint $(2, \Phi_1)$, corresponding to the two vertices Φ_1, Φ_2 of the uncertainty polytope. This *branching effect*, which becomes more frequent with increasing state dimension, is the primary reason why

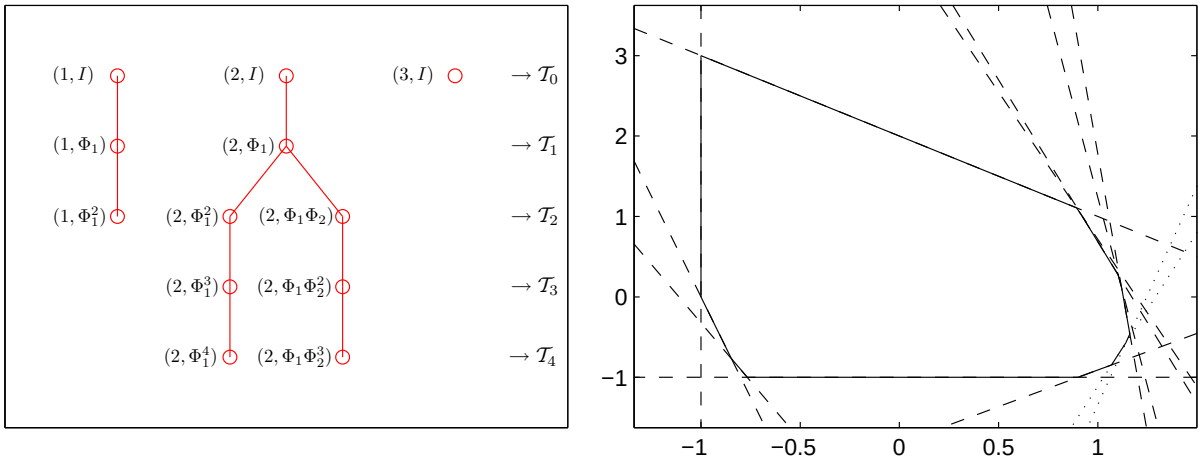

Fig. 1. Illustration of the structure of the Maximal Admissible Set for an example system defined by $n_x = 2, L = 2, \Phi_1 = [1\ 0.2;\ -0.5\ 0.55]$ and $\Phi_2 = [1\ 0.2;\ -0.55\ 0.45]$, subject to constraints $A_x x \leq 1_{\mathrm{v}}$ with $A_x = [-1\ 0; 0\ -1; 0.5\ 0.5]$. **Left:** the tree structure of the MAS, where the nodes of the tree represent the different constraints of the MAS, with the constraints of the different sets $\mathcal{T}_i$ appearing at different depths of the trees as indicated on the figure. The notation (i, M) denotes the constraint $a_i^{\mathrm{T}} M x \leq 1$, with a_i^{T} denoting the i-th row of A_x. **Right:** a depiction of the MAS. The constraints $(1, *)$ bound the MAS from the left; constraints $(2, *)$ bound the MAS from the lower right; constraint $(3, I)$ is the diagonal constraint bounding the MAS from above. Constraints $(2, \Phi_1^2)$ and $(2, \Phi_1\Phi_2)$ are depicted as dotted lines.

polyhedral invariant sets for LPV systems can be dramatically more complex than invariant sets for LTI systems. Therefore, the main purpose of this section is to find a way to efficiently reduce this branching effect.

Eliminating the branching effect observed in Figure 1 is possible by tightening one of the two constraints $(2, \Phi_1^2)$ and $(2, \Phi_1\Phi_2)$. When comparing the two dotted lines it is clear that by tightening either of both constraints with only a small factor, the other becomes redundant. Furthermore, as indicated by Remark 6, choosing the sets $\mathcal{T}_i$ slightly smaller than necessary in step 2b) of Algorithm 2 does not violate (14) after termination of the algorithm. Constraint tightening can therefore be considered to be a viable way to reduce the number of branches without losing invariance of the resulting set. The problem of finding how much a constraint should be tightened in order to make another constraint redundant can be formulated as follows :

Problem 3 (P3) *Given real matrices $A \in \mathbb{R}^{m\times n}, a_1 \in \mathbb{R}^n, a_2 \in \mathbb{R}^n$ with $a^\star > 1$,*

$$\nu = \max_x\ a_2^{\mathrm{T}} x \quad \text{s.t.} \quad \begin{bmatrix} A \\ a_1^{\mathrm{T}} \end{bmatrix} x \leq 1_{\mathrm{v}}, \tag{16}$$

find a scalar $\eta > 1$ such that $\nu' = 1$, with

$$\nu' = \max_x\ a_2^{\mathrm{T}} x \quad \text{s.t.} \quad \begin{bmatrix} A \\ \eta a_1^{\mathrm{T}} \end{bmatrix} x \leq 1_{\mathrm{v}}. \tag{17}$$

Remark 7 *The property $\nu > 1$ indicates that $a_2^{\mathrm{T}} x \leq 1$ in not redundant with respect to $[A;\ a_1^{\mathrm{T}}] x \leq 1_{\mathrm{v}}$. The scalar η is the factor with which $a_1^{\mathrm{T}} x \leq 1$ should be made stricter in order to make $a_2^{\mathrm{T}} x \leq 1$ redundant.*

The following lemma allows Problem P3 to be solved by solving a single LP :

Lemma 2 *Given Problem P3, and given the optimal solution $x^{\mathrm{o}}, \kappa^{\mathrm{o}}$ of the following optimization problem*

$$\min_{x,\kappa}\ \kappa \quad \text{s.t.} \quad \begin{bmatrix} A & 0 \\ a_1^{\mathrm{T}} & -1 \end{bmatrix} \begin{bmatrix} x \\ \kappa \end{bmatrix} \leq \begin{bmatrix} 1_{\mathrm{v}} \\ 0 \end{bmatrix},\ a_2^{\mathrm{T}} x = 1, \tag{18}$$

then $\eta = \frac{1}{\kappa^{\mathrm{o}}}$ is a solution to P3 if $\kappa^{\mathrm{o}} > 0$.

Proof : Due to the assumption of P3 that $\nu > 1$ it is guaranteed that there exists a vector x^{f} such that $[x;\ \kappa] = [x^{\mathrm{f}};\ 1]$ is strictly feasible for (18) and hence $\kappa^{\mathrm{o}} < 1$. by rewriting the constraints of (18) into the form of (17) it is clear that x^{o} is a feasible solution to (17) for $\eta = \frac{1}{\kappa^{\mathrm{o}}}$ if $\kappa^{\mathrm{o}} > 0$. Furthermore, by writing the optimality conditions of (18), it can be seen that x^{o} also is an optimal solution to (17) for $\eta = \frac{1}{\kappa^{\mathrm{o}}}$. Due to the constraint $a_2^{\mathrm{T}} x = 1$ of (18) it is clear that $\nu' = 1$, which proves correctness of the lemma if $\kappa^{\mathrm{o}} > 0$. □

The above lemma provides an efficient method for checking whether tightening a given constraint, can eliminate a branch in the constraint tree. In the following sections this method will be referred to as *pruning*.

4.2 Algorithm formulation using pruning

In this section constraint tightening is incorporated into Algorithm 1 in order to construct invariant sets with a

reduced number of constraints compared to the MAS constructed using Algorithm 1. For simplicity the Algorithm will only be given for the case $L = 2$, but extensions to $L > 2$ are straightforward. For future reference intermediate results of the algorithm will be denoted as $\mathcal{V}_0, \mathcal{V}_{i,\mathrm{a}}, \ldots, \mathcal{V}_{i,\mathrm{c}}, i = 1, \ldots, n$.

Algorithm 3 (Algorithm using pruning) *Consider an LPV system* (1)-(2) *subject to constraints* (3) *and satisfying the condition* $\max_j \|\Phi_j\| < 1$. *Furthermore, consider the user-specified scalar parameters* $d_1 > \max_j \|(A_x)_{[j,:]}\|$, $d_2 \in [\max_j \|\Phi_j\|, 1)$ *and* $\gamma > 0$. *Perform the following operations :*

(1) Initialize $\mathcal{T}_0 := \mathcal{X}$. *Set* $\mathcal{V}_0 := \mathcal{T}_0$.
(2) Initialize $i := 1$.
(3) Iterate until $\mathcal{S}_{i-1} \subseteq \mathcal{T}_{i-1}^-$ *:*
  (a) Set $\mathcal{T}_i := \mathcal{T}_{i-1}^-$. *Set* $\mathcal{V}_{i,\mathrm{a}} := \bigcap_{k=0}^{i} \mathcal{T}_k$.
  (b) Remove all constraints from $\mathcal{T}_{i-1}$ *and* $\mathcal{T}_i$ *that are redundant with respect to* $\mathcal{V}_{i,\mathrm{a}}$. *Set* $\mathcal{V}_{i,\mathrm{b}} := \bigcap_{k=0}^{i} \mathcal{T}_k$.
  (c) Check whether any branching occurred when constructing set $\mathcal{T}_i$ *and perform for each such constraint pair (denoted as* $(T_i)_{[j_1,:]}$ *and* $(T_i)_{[j_2,:]}$*) the following steps :*
    (i) Use Lemma 2 to calculate the scalars $\eta_1 > 1, \eta_2 > 1$ *(if both exist) needed to make one of the constraints redundant by tightening the other. Assume without lack of generality that* $\eta_1 < \eta_2$.
    (ii) If $\eta_1 \leq 1 + \gamma$ *and* $\eta_1 \|(T_i)_{[j_1,:]}\| \leq d_1 \cdot d_2^i$, *then set* $(T_i)_{[j_1,:]} := \eta_1 (T_i)_{[j_1,:]}$.
  (d) Set $\mathcal{V}_{i,\mathrm{c}} := \bigcap_{k=0}^{i} \mathcal{T}_i$ *and set* $i := i + 1$.

Return the set $\mathcal{S} = \mathcal{V}_{i,\mathrm{c}}$ *and set* $n := i$.

Remark 8 *The parameters* d_1 *and* d_2 *have the purpose of guaranteeing convergence of the algorithm, as will become clear later in this section. The parameter* γ *defines an upper bound to the tightening factors applied in the algorithm and therefore influences the trade-off between the amount of pruning and the volume of the resulting invariant set.*

The following theorems show that the above algorithm terminates for a finite value of n (i.e. convergence) and that the resulting set $\mathcal{S}$ is a feasible invariant set with respect to (1)-(2),(3) (i.e. correctness).

Theorem 1 (Convergence) *Algorithm 3 terminates in a finite number of iterations.*

Proof : We prove by induction that in each iteration after step 3c) the following condition is satisfied :

$$\max_j \|(T_i)_{[j,:]}\| \leq d_1 \cdot d_2^i. \tag{19}$$

As a result, for sufficiently large i, all constraints from $\mathcal{T}_i^-$ will be redundant with respect to $\mathcal{S}_i$, after which the algorithm will terminate.

Due to the choice of d_1 it can be verified that after step 1) condition (19) is satisfied for $i = 0$. The removal of rows from a matrix T_n can only decrease the value of $\max_j \|(T_n)_{[j,:]}\|$, so condition (19) is also still satisfied for $n = 0$ after step 2). Assume (19) holds for n, then it will also hold for $n + 1$ after step 4a), due to the choice of d_2. This property still holds in the next iteration after step 4b), since removal of constraints can only decrease $\max_j \|(T_i)_{[j,:]}\|$. Step 4cii) explicitly prohibits constraint tightening to take place if (19) would become violated. Therefore, this condition is always valid for n after step 4c), which therefore leads to the termination of the algorithm for a finite value of n by virtue of the above arguments, which in turn proves the lemma. □

Theorem 2 (Correctness) *The set* $\mathcal{S}$ *resulting from Algorithm 3 is a feasible invariant set with respect to* (1)-(2),(3).

Proof : First of all it should be noted that the set $\mathcal{S}_i$ only decreases in size as a function of i after each iteration, since constraints are only added or tightened and non-redundant constraints are never removed. Therefore, due to the initialization of $\mathcal{T}_0$ it is clear that $\mathcal{S} \subseteq \mathcal{X}$, which proves feasibility.

In order to prove invariance we have to show that $\mathcal{S} \subseteq \mathcal{S}^-$, or, reformulated to take the structure of $\mathcal{S}$ into account :

$$\mathcal{S} \subseteq \mathcal{T}_i^-, \qquad i = 0, \ldots, n. \tag{14}$$

Since removal of redundant constraints does not modify $\mathcal{S}$ and can only increase the size of the sets $\mathcal{T}_i^-$, it is clear that step 3b) can be ignored in this proof.

It is clear that after step 3a) the condition $\mathcal{V}_{i,\mathrm{a}} \subseteq \mathcal{T}_{i-1}^-$ is satisfied. Also, one can see that $\mathcal{V}_{i,\mathrm{c}} \subseteq \mathcal{V}_{i,\mathrm{b}} = \mathcal{V}_{i,\mathrm{a}}$ and therefore it is also guaranteed that $\mathcal{V}_{i,\mathrm{c}} \subseteq \mathcal{T}_{i-1}^-$. Since $\mathcal{V}_{i+1,\mathrm{a}} \subseteq \mathcal{V}_{i,\mathrm{c}}$ it is proven by induction that also $\mathcal{V}_{j,\mathrm{c}} \subseteq \mathcal{T}_{i-1}^-, \forall j \geq i$. This shows that $\mathcal{S} \subseteq \mathcal{T}_i^-, i = 0, \ldots, n-1$. Finally, due to the termination condition it is also guaranteed that $\mathcal{S} \subseteq \mathcal{T}_n^-$ which then proves invariance of $\mathcal{S}$. □

Figure 2 shows how Algorithm 3 is able to construct a reduced complexity invariant set for the system depicted in Figure 1.

4.3 Parameter tuning

Parameters d_1 and d_2 are primarily used to impose convergence of the algorithm and can mostly given the fol-

lowing default values :

$$d_1 = d \cdot \max_i \|(A_x)_{[i,:]}\|, \tag{20a}$$

$$d_2 = \max_j \|\Phi_j\|, \tag{20b}$$

with $d \approx 1.5$. An interesting property that can easily be proven is that the maximum volume reduction to be expected is proportional to d^{n_x}, where n_x is the state dimension of the system. Therefore, in cases where large volume reductions can be tolerated, larger values of d can be chosen.

The parameter $\gamma \geq 0$ is the main parameter influencing the trade-off between low complexity and high volume since it defines an upper bound on the factor with which constraints can be tightened. Larger values of γ lead to lower complexity invariant sets, while smaller values for γ lead to larger invariant sets. A good starting point usually is $\gamma \approx 0.1$. Systems with a larger amount of uncertainty typically require larger values for γ to obtain equally large complexity reductions.

5 Tree depth reduction : invariant set construction using *trimming*

In this section a new theorem is introduced that adds an additional tuning parameter to the algorithm for constructing reduced complexity invariant sets. It is shown that, given an LPV system, it is allowed to modify the system and calculate the invariant set for the modified system. Under certain conditions the invariant set for the modified system is also invariant with respect to the original system. The system modification process involves a parameter that is freely chosen by the user, giving additional flexibility that can potentially result in invariant sets with shorter constraint trees. In the following sections this method will be referred to as *trimming*.

Theorem 3 (Sets for Modified System Matrices) *Given a set $\mathcal{S} \in \mathbb{R}^{n_x}$ and two autonomous LPV systems defined by uncertainty polytopes Ω_1 and Ω_2 that are defined as*

$$\Omega_1 \equiv \mathrm{Co}(\Phi_1, \ldots, \Phi_L), \tag{21a}$$

$$\Omega_2 \equiv \mathrm{Co}(\Phi'_1, \ldots, \Phi'_L), \tag{21b}$$

with

$$\Phi'_i = (1+c)\Phi_i - cI, \qquad i = 1, \ldots, L, \tag{22}$$

where $c \in \mathbb{R}^+$. If $\mathcal{S}$ is convex and invariant with respect to Ω_2, then it is also invariant with respect to Ω_1.

Proof : If $\mathcal{S}$ is invariant with respect to Ω_2 then

$$\Phi'_i x \in \mathcal{S}, \qquad \forall x \in \mathcal{S}, i = 1, \ldots, L,$$

or equivalently

$$((1+c)\Phi_i x - cx) \in \mathcal{S}, \qquad \forall x \in \mathcal{S}, i = 1, \ldots, L.$$

Due to convexity of $\mathcal{S}$ and the fact that $c > 0$ it is therefore also guaranteed that

$$\left(\frac{1}{1+c}((1+c)\Phi_i x - cx) + \frac{c}{1+c}x\right) \in \mathcal{S}, \qquad \forall x \in \mathcal{S}, i = 1, \ldots, L,$$

which, after some algebraic manipulation, is equivalent to

$$\Phi_i x \in \mathcal{S}, \qquad \forall x \in \mathcal{S}, i = 1, \ldots, L,$$

which proves that $\mathcal{S}$ is invariant with respect to the LPV system defined by Ω_1. □

The above theorem indicates that, given system defined by uncertainty polytope Ω_1, one can calculate an invariant set $\mathcal{S}$ for the system defined by Ω_2, which is then invariant for both systems. The set $\mathcal{S}$ will be a subset of the MAS for the original system. The parameter c can be used to reduce the number of constraints of the resulting set $\mathcal{S}$.

The theorem can be further generalized by, among other things, allowing different values of c for each Φ_i-matrix.

5.1 Parameter tuning

The above described method for computing reduced complexity invariant sets based on modified system matrices is especially useful for systems where the eigenvalues of the Φ_i matrices lie close to the unity point $1 + 0i$, which is often the case in real-life examples. Invariant sets for such systems typically have a large number of constraints due to the large values of n required in these cases to satisfy (14). Choosing $c > 0$ will push these eigenvalues away from the unity point and can therefore potentially reduce the number of constraints of the obtained invariant set $\mathcal{S}$. Choosing too large a value for c will result in eigenvalues outside the unit circle, in which case no invariant set can be found for the modified system. Therefore, based upon the eigenvalues $\alpha_k + \beta_k j$ of the matrices Φ_i, one can determine an interval within which the parameter c should be chosen :

$$c \in \left[0, \min_k \left(\frac{2}{\sqrt{(1-\alpha_k)^2 + \beta_k^2}}\right) - 1\right]. \tag{23}$$

Values of c outside this interval will lead to unstable systems.

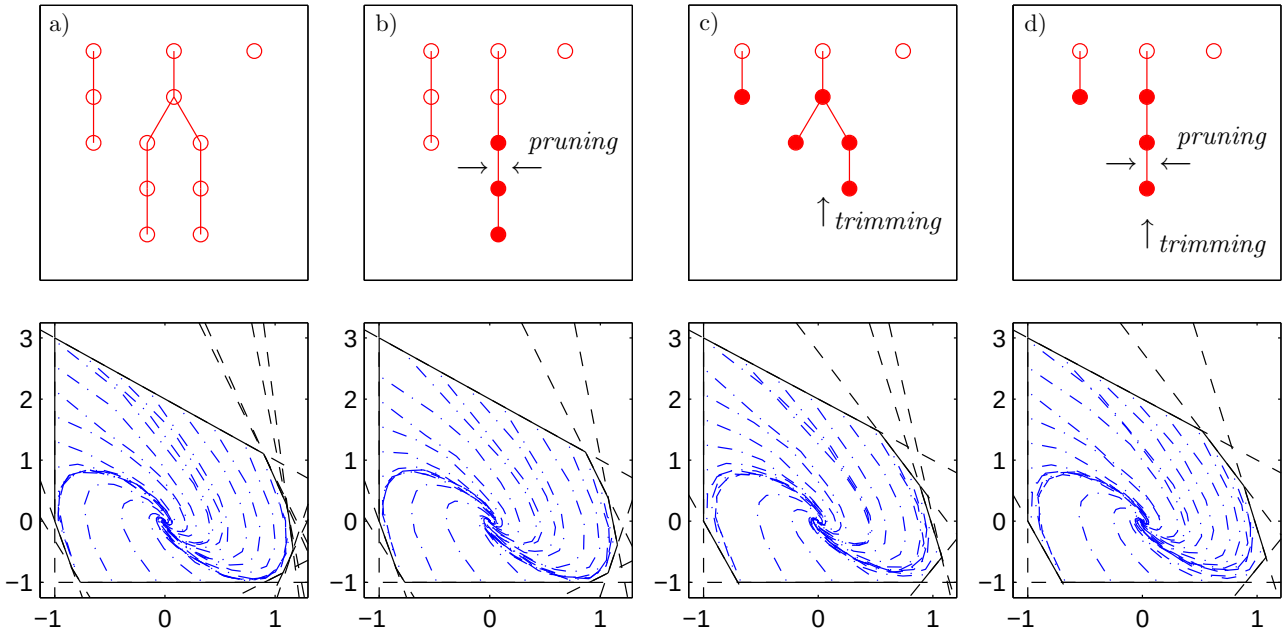

Fig. 2. Depiction of the results of pruning and trimming when applied to the system discussed in Figure 1. **Top :** Constraint trees of the resulting invariant sets. Solid circles depict constraints that are modified with respect to the MAS (i.e. subfigure a)). **Bottom :** Geometric depiction of the resulting invariant sets along with trajectories starting near the boundaries of the sets. **Left to right : a)** the MAS of the system, **b)** invariant set obtained using *pruning* (cfr. Algorithm 3) with parameters $d_1 = 1.5 \max_i \|(A_x)_{[i,:]}\|, d_2 = \max_j \|\Phi_j\|, \gamma = 0.1$, **c)** the MAS obtained using *trimming* (cfr. Section 5) with $c = 0.5$, **d)** result obtained using both pruning and trimming using identical parameters as in subfigures b) and c).

6 Examples

Three sets of examples are provided. The first subsection shows examples illustrating the use of constraint tightening in order to construct reduced complexity invariant sets. The second subsection shows how reduced complexity invariant sets can be constructed based on modified systems. The third subsection finally illustrates the performance of both methods when applied to a specific class of autonomous LPV systems encountered in robust MPC algorithms.

A MATLAB implementation of Algorithm 3 that only differs in some minor implementational aspects can be found at the File Exchange section of the MATLAB Central [1] . This code was used for the examples shown in this section.

6.1 Pruning

For a first example we refer to Figure 2 which shows an invariant set for the same system as the one depicted in Figure 1. Algorithm 3 is used to construct the set and is able to eliminate one of the branches by means of pruning.

[1] http://www.mathworks.com/matlabcentral/fileexchange

To give a better idea of the performance of Algorithm 3, we perform Monte Carlo studies on randomly generated LPV models of the following form :

$$\Phi_1 = c_1(I + 0.1R_1), \tag{24a}$$
$$\Phi_2 = \Phi_1 + c_2 R_2, \tag{24b}$$

where $R_1, R_2 \in \mathbb{R}^{n_x \times n_x}$ are matrices with random entries between -1 and $+1$. The scalars c_1, c_2 are chosen such that $\|\Phi_1\| = 0.99$ and $\|c_2 R_2\| = 0.01$. The systems are subject to constraints $[-I; 1_{\mathrm{v}}^{\mathrm{T}}]x \leq 1_{\mathrm{v}}$. Such models can be considered to be representative for slow systems that are operated near their operating constraints (e.g. chemical plants), which is where constrained control is typically employed and set invariance can be a useful tool.

Table 1 shows how the algorithms perform for different state dimensions $n_x = 2, \ldots, 10$. Computational complexities of the algorithms are expressed in terms of the average number of Linear Programs (LP's) to be solved per invariant set and the average number of constraints per LP. It is clear that Algorithm 3 is able to reduce the number of constraints defining the resulting invariant set with increasing efficiency as the state dimension grows. It should be noted that only a negligible volume reduction was observed for these artificial examples. We refer to Section 6.3 for an example where the trade-off between volume and complexity is better observable. Al-

n_x	rows($A_{\mathcal{S}}$)	#LP's	size(LP's)
2	32.9 (37.4)	248.4 (284.6)	25.8 (31.2)
3	40.8 (99.5)	340.1 (733.4)	28.7 (100.4)
4	56.7 (409.0)	531.2 (2994.9)	39.4 (772.8)
5	65.6 (-)	630.8 (-)	46.5 (-)
10	120.4 (-)	1229.2 (-)	83.9 (-)

Table 1
Results of a Monte Carlo simulation on 100 random systems of the form (24) using Algorithm 3 with parameters $d_1 = 1.5 \max_i \|(A_x)_{[i,:]}\|, d_2 = \max_j \|\Phi_j\|, \gamma = 0.1$ and for $n_x = 1, \ldots, 10$. The reported numbers are averages and denote (from left to right) the number of constraints describing the invariant sets, the volume of a 2-dimensional intersection of the invariant set along the first 2 dimensions, the number of LP's to be solved in order to construct the invariant set and the number of inequality constraints per LP. The numbers between brackets are the ones obtained for Algorithm 2.

though constraint tightening seems to induce an additional computational load, this study indicates that the total computational complexity is also significantly reduced

6.2 Trimming

In this section we repeat the Monte Carlo study presented in Table 1, but the invariant sets are calculated using trimming as discussed in Section 5. A value of $c = 0.5$ is chosen. It is clear that the number of constraints is further reduced.

6.3 Invariant sets for robust MPC

In this section we test the new methods on the invariant set problems explained in [14]. This involves the construction of invariant sets for an autonomous LPV system of the following form :

$$\Phi_i = \begin{bmatrix} A_i - B_i K & B_i & 0 \\ 0 & 0 & I_{(N-1)\cdot n_u} \\ 0 & 0 & 0 \end{bmatrix}, \qquad i = 1, 2, \quad (25)$$

n_x	rows($A_{\mathcal{S}}$)	#LP's	size(LP's)
2	23.0	168.6	18.9
3	28.8	229.8	21.3
4	40.0	356.3	29.2
5	46.7	421.7	34.7
10	85.1	831.5	63.9

Table 2
Results of a Monte Carlo simulation identical to the one presented in Table 1 but system modification (cfr. Section 5) with $c = 0.5$ is performed before calculating the invariant sets. We refer to Table 1 for more details.

for increasing values of $N > 1$, where $I_n \in \mathbb{R}^{n \times n}$ denotes a unity matrix. The resulting invariant set is used to characterize the allowable control moves for the constrained controller and therefore, the volume of the invariant set is of main importance. More specifically, the projection on the first n_x (in this case 2) dimensions gives the region within which the controller is well-defined. We will denote the volume of this projection as V.

On the other hand the number of constraints describing the invariant set is a determining factor in the online computational efficiency of the controller and therefore this number is preferably kept low. Typically, both V and the number of constraints will increase with increasing N. We refer to [14,16] for more details.

Numerical values of A_i, B_i are chosen as :

$$A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 1 & 0.11 \\ 0 & 1 \end{bmatrix}, \quad (26)$$

$$B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 \\ 1.1 \end{bmatrix}, \quad (27)$$

while K is defined as [0.5 0.3]. Constraints are defined as

$$\begin{bmatrix} -10 \cdot 1_{\mathrm{v}} \\ -1 \end{bmatrix} \leq \begin{bmatrix} I_2 & 0 & 0 \\ -K & 1 & 0 \end{bmatrix} x \leq \begin{bmatrix} 10 \cdot 1_{\mathrm{v}} \\ 0.5 \end{bmatrix}. \quad (28)$$

Pruning and trimming is applied for various values of N and γ. Table 3 reports V and the number of constraints describing the obtained invariant sets as a function of N. It can be seen that the complexity of the MAS increases exponentially with N, whereas the number of constraints describing the invariant sets constructed using pruning seems to increase more or less as a linear function of N, allowing the use of larger values of N. In this way larger volumes V can be obtained with a smaller number of constraints, which justifies the volume reduction observed for fixed values of N. The influence of trimming is also clear in this example. The number of constraints is further reduced, but at the cost of a further reduction of the volume. For lower-dimensional systems the effect of trimming seems to be less effective in this case, since the number of constraints increases, contrary to what is expected. A more detailed analysis of the interaction between pruning and trimming and the choice of γ and c forms the subject of further research.

Figure 3 depicts the number of constraints and the volume of invariant sets obtained using Algorithm 3 for $N = 5$ and various values of γ. Increasing values of γ lead to larger reductions in both the number of constraints as the volume V. However, the volume reduction can be

	$c = 0$		$c = 0.1$	
N	rows($A_{\mathcal{S}}$)	V	rows($A_{\mathcal{S}}$)	V
1	18 (28)	46.5 (47.7)	19 (45)	45.3 (46.5)
2	24 (55)	59.1 (62.4)	28 (93)	57.3 (60.4)
3	32 (99)	70.8 (76.7)	44 (171)	69.9 (74.0)
4	61 (179)	85.6 (91.9)	63 (301)	82.6 (88.1)
5	74 (326)	100.1 (107.9)	66 (-)	96.1 (-)
10	107 (-)	160.1 (-)	97 (-)	149.8 (-)
25	195 (-)	302.9 (-)	188 (-)	289.0 (-)

Table 3
Number of constraints and volume V of invariant sets for system (25)-(28) computed with Algorithm 2 (between brackets) and Algorithm 3 for different values of N. The parameters were chosen as $d_1 = 5 \max_i \|(A_x)_{[i,:]}\|, d_2 = \max_j \|\Phi_j\|, \gamma = 0.05$.

seen to be much more modest. It should be noted however, that this trade-off also depends on the size of the uncertainty polytope Ω.

7 Conclusion

This paper deals with the construction of polyhedral invariant sets for LPV systems with polytopic uncertainty description that make a trade-off between the complexity of the invariant set (expressed in terms of the number of constraints describing the set) and its volume. Two main contributions can be distinguished. The first contribution is the description of the structure of the Maximal Admissible Set (MAS) for LPV systems and the analysis of how this structure can be exploited to decrease the complexity of the resulting sets. The second contribution is the formulation of two efficient methods (*pruning* and *trimming*) based on the insights of the first contribution that allow the construction of lower-complexity invariant inner approximations of the MAS. The new algorithms also exhibit a computational complexity that can be orders of magnitude lower than the time needed to construct the MAS. Finally, numerical examples show the efficiency of the new methods on a variety of examples.

There are several possible directions for future research. The first is the further integration of the two methods described in this paper and a more detailed analysis of the interaction between the different tuning parameters. A second possible research direction is the construction of invariant sets for LPV systems subject to bounded disturbances. A third interesting research direction is the analysis of the performance of constrained control techniques when based on the reduced complexity invariant sets discussed in this paper.

Acknowledgements

Research supported by KUL: GOA-Mefisto 666, GOA-AmbioRics; FWO: G.0240.99, G.0407.02, G.0197.02, G.0141.03, G.0491.03, G.0120.03, G.0800.01, G.0452.04, G.0499.04, G.0211.05, G.0080.01, G.0226.06, research communities (ICCoS, ANMMM); IWT: PhD Grants, BFSPO: IUAP P5/22; PODO-II (CP/40: TMS and Sustainability); EU: FP5-CAGE; FP5-Quprodis; ERNSI; FP6-BioPattern; Eureka 2419-FliTE; Contract Research/agreements: Data4s, Electrabel, Elia, LMS, IPCOS, VIB; Bert Pluymers is a research assistant with the I.W.T. at the Katholieke Universiteit Leuven, Belgium. Dr. Johan Suykens is an associate professor and Dr. Bart De Moor is a full professor at the Katholieke Universiteit Leuven, Belgium

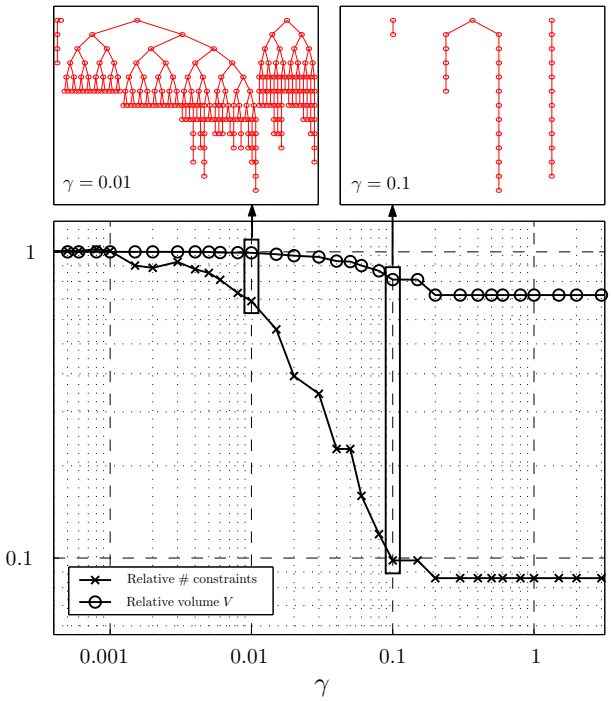

Fig. 3. Relative number of constraints and relative volume V (both with respect to the MAS, i.e $\gamma = 0$) of invariant sets for system (25)-(28) with $N = 5$ computed with Algorithm 3 for different values of γ. The other parameters were chosen as $d_1 = 5 \max_i \|(A_x)_{[i,:]}\|, d_2 = \max_j \|\Phi_j\|$. The constraint structures of the invariant sets corresponding to $\gamma = 0.01$ and $\gamma = 0.1$ are depicted at the top of the figure.

References

[1] F. Blanchini. Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions. *IEEE Transactions on Automatic Control*, 39:428–433, 1994.

[2] F. Blanchini. Set invariance in control. *Automatica*, 35:1747–1767, 1999.

[3] F. Blanchini, S. Miani, and C. Savorgnan. Polyhedral lyapunov functions computation for robust and gain scheduled design. *Proceedings of the Symposium on nonlinear Control Systems (NOLCOS) 2004, Stuttgart, Germany.*

[4] M. Cannon, V. Deshmukh, and B. Kouvaritakis. Nonlinear model predictive control with polytopic invariant sets. *Automatica*, 39:487–494, 2003.

[5] M. Cannon, B. Kouvaritakis, and V. Deshmukh. Enlargement of polytopic terminal region in nmpc by interpolation and partial invariance. *Automatica*, 2004.

[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 Transactions on Automatic Control*, 36(9):1008–1020, 1991.

[7] E. Kerrigan. *Robust Constraint Satisfaction: Invariant Sets and Predictive Control*. PhD thesis, Cambridge, 2000.

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

[9] W. Langson, I. Chryssochoos, S.V. Raković, and D.Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.

[10] Y. I. Lee and B. Kouvaritakis. Robust receding horizon predictive control for systems with uncertain dynamics and input saturation. *Automatica*, 36:1497–1504, 2000.

[11] 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.

[12] B. Pluymers, L. Roobrouck, J. Buijs, J. A. K. Suykens, and B. De Moor. Model-predictive control with time-varying terminal cost using convex combinations. *Automatica*, 41:831–837, 2005.

[13] 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 description. *Proceedings of the American Control Conference (ACC) 2005, Portland, USA*.

[14] B. Pluymers, J.A. Rossiter, J.A.K. Suykens, and B. De Moor. Efficient computation of polyhedral invariant sets for LPV systems and application to robust MPC. *Submitted for publication*, 2005, (http://www.esat.kuleuven.ac.be/~sistawww/cgi-bin/pub.pl).

[15] J.A. Rossiter. *Model Based Predictive Control*. CRC Press, 2003.

[16] J.A. Rossiter, B. Kouvaritakis, and M.J. Rice. A numerically robust state-space approach to stable predictive control strategies. *Automatica*, 34:65–73, 1998.

[17] P.O.M. Scokaert and J.B. Rawlings. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, 43(8):1163–1168, 1998.