Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 05-133

Efficient Computation of Polyhedral Invariant Sets for LPV Systems and Application to Robust MPC [1]

B. Pluymers[2] , J.A. Rossiter[3] , J. Suykens[2] and B. De Moor[2]

May 2005

Submitted for publication

[1]This report is available by anonymous ftp from *ftp.esat.kuleuven.ac.be* in the directory *pub/sista/pluymers/reports/IEEE-TAC05_RobInv.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.ac.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.

[3]University of Sheffield, Department of Automatic Control and Systems Engineering, Mappin Street, Sheffield S1 3JD, United Kingdom, E-mail: *j.a.rossiter@sheffield.ac.uk*

Abstract

Recent results have made extensive use of invariant sets in order to guarantee recursive feasibility and stability in Model Based Predictive Control (MPC) algorithms. When dealing with linear time-invariant (LTI) systems the use of polyhedral invariant sets is preferable, compared to say ellipsoidal invariant sets, due to the increased size of the set of feasible initial conditions and the fact that the use of linear constraints in the on-line optimization problems allows Quadratic Programming (QP) based formulations. However, when dealing with linear parameter-varying (LPV) systems, polyhedral invariant sets have remained largely unused, except in rather restricted forms due to the lack of computationally feasible synthesis algorithms. In this paper the construction of the maximal size polyhedral invariant set (known as the Maximal output Admissible Set or MAS), introduced by Gilbert *et al.* for LTI systems, is extended to LPV systems with polytopic uncertainty description. The algorithm is based on a new invariance condition allowing the intermediate removal of redundant constraints, thereby dramatically reducing the number of constraints describing the invariant set. The polyhedral robust invariant sets arising are used to formulate a robust quasi-infinite horizon MPC algorithm requiring only a QP online but with significantly improved feasibility properties. Stability and recursive feasibility are proven rigorously and the efficacy is demonstrated by means of a numerical example.

Efficient Computation of Polyhedral Invariant Sets for LPV Systems and Application to Robust MPC

B. Pluymers†, J.A. Rossiter‡, J.A.K. Suykens†, B. De Moor†

†Katholieke Universiteit Leuven
Department of Electrical Engineering, ESAT-SCD-SISTA
E-Mail : {bert.pluymers, johan.suykens, bart.demoor}@esat.kuleuven.ac.be
Internet : `http://www.esat.kuleuven.ac.be/scd/`

‡University of Sheffield
Department of Automatic Control and Systems Engineering
E-Mail : j.a.rossiter@sheffield.ac.uk
Internet : `http://www.shef.ac.uk/acse/`

Monday 30th May, 2005

Abstract

Recent results [9,11] have made extensive use of invariant sets [2] in order to guarantee recursive feasibility and stability in Model Based Predictive Control (MPC) algorithms. When dealing with linear time-invariant (LTI) systems the use of polyhedral invariant sets is preferable, compared to say ellipsoidal invariant sets, due to the increased size of the set of feasible initial conditions and the fact that the use of linear constraints in the on-line optimization problems allows Quadratic Programming (QP) based formulations. However, when dealing with linear parameter-varying (LPV) systems, polyhedral invariant sets have remained largely unused, except in rather restricted forms [6, 11] due to the lack of computationally feasible synthesis algorithms. In this paper the construction of the maximal size polyhedral invariant set (known as the Maximal output Admissible Set or MAS), introduced by Gilbert *et al.* [7] for LTI systems, is extended to LPV systems with polytopic uncertainty description. The algorithm is based on a new invariance condition allowing the intermediate removal of redundant constraints, thereby dramatically reducing the number of constraints describing the invariant set. The polyhedral robust invariant sets arising are used to formulate a robust quasi-infinite horizon MPC algorithm requiring only a QP online but with significantly improved feasibility properties. Stability and recursive feasibility are proven rigorously and the efficacy is demonstrated by means of a numerical example.

1 Introduction

Model based Predictive Control (MPC) is a control paradigm that at each sample instant solves a finite-horizon optimal control problem in a receding horizon fashion. The main advantage of MPC is the ability to deal explicitly with constraints but, in its traditional form, it has no guarantee

for stability or recursive feasibility [14]. To tackle this problem, several authors proposed the implicit use of a linear terminal feedback controller beyond the prediction horizon. By adding the 'terminal' constraint that the last state within the prediction horizon (terminal state) should lie within an invariant set associated with this terminal controller, recursive feasibility can be proven. However, this terminal constraint has important implications on the on-line optimization problem in that: a) the size of the invariant set influences the set of initial states for which the optimization problem is feasible (feasible region) and b) the description of the invariant set (linear, quadratic, ..., inequalities) determines the optimization class (QP, SDP, ...) and hence also the on-line computational efficiency of the associated MPC algorithm. The aim of this paper is to construct polyhedral invariant sets for LPV systems and then form corresponding robust MPC algorithms which improve feasibility over existing approaches and moreover require the use, on-line, of only QP optimisers.

The notion of set invariance [2, 7, 8] is essential to this paper and also arises in many other problems concerning analysis of dynamical systems, controller design and the construction of Lyapunov functions. A systematic way for constructing maximal volume polyhedral sets (Maximal output Admissible Set, MAS) for linear time-invariant (LTI) systems was initially proposed in [7]. The proposed algorithm constructs an invariant set by iteratively adding additional constraints until invariance is obtained. However, constructing the MAS for linear parameter-varying (LPV) systems is a problem for which no efficient algorithms have been published, although a number of contributions in this direction are known.

In [1] the construction of controllability sets for linear systems with polytopic model uncertainty and polytopic disturbances is described. These sets do not take a given controller into account, but rather guarantee that for each state inside the set, some control action exists that steers the system further inside the set with a given convergence rate. In [8] and related works, theoretical results related to invariant sets for uncertain systems with disturbances are discussed, but no general algorithms for the setting considered in this paper have been proposed. In [11] a method

is proposed to construct low-complexity robust invariant sets for uncertain linear systems driven by a linear feedback controller. A set defined by component-wise bounds in a similarly transformed state space is considered and invariance is imposed by demanding that the Perron-Frobenius norm of the closed loop system matrices is smaller than unity. However, this leads to conservative invariant sets and in some cases no invariant set can be obtained. Another, but also conservative, approach is the construction of ellipsoidal invariant sets; this has the advantage that it can be formulated as a single convex optimization problem [4, 9, 13].

The main contribution of this paper is the introduction of an efficient algorithm that constructs the MAS for LPV systems with polytopic model uncertainty subject to linear constraints. An invariance condition more general than that discussed in [7, Theorem 2.2] is proposed, leading to an increased efficiency of our algorithm compared to [7, Algorithm 3.2] and enabling the extension towards LPV systems with polytopic uncertainty. Extensions taking bounded disturbances and imposed convergence rates into account are also discussed briefly.

The second contribution is the formulation of an MPC algorithm for LPV systems making use of the proposed MAS. The new algorithm is essentially an extension of work in [10, 18, 20], which relies on ellipsoidal robust invariant sets. The algorithm proposed here has the advantages: a) the feasible region is significantly larger than that of the ellipsoid based algorithm and b) the on-line optimization problem can be cast as a quadratic program (QP) instead of an semi-definite program (SDP) and hence can be solved more efficiently. Robust stability and recursive feasibility of the algorithm are proven rigorously.

The paper is organized as follows. In Section 2 the problem of constructing the MAS for LPV systems is formulated, after which, in Section 3, a theoretical approach is taken towards the solution to this problem. In Section 4 a generalized invariance condition is formulated leading to an efficient algorithm for constructing the MAS. Section 5 introduces a new polyhedral invariant set based robust MPC algorithm. Numerical examples appear in Section 6 and Section 7 states the conclusions and future work.

2 Set Invariance

In the following sections we consider LPV systems of the form

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

$$y(k) = Cx(k), \tag{1b}$$

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

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

The output is subject to linear constraints

$$y(k) \in \mathcal{Y} = \{y | A_y y \leq b_y\}, \qquad k = 0, \ldots, \infty, \tag{3}$$

with $0 \in \mathcal{Y}$, which is equivalent with $b_y \geq 0$. In the sequel, we translate the output constraints $\mathcal{Y}$ into state constraints when appropriate :

$$x(k) \in \mathcal{X} = \{x | A_x x \leq b_x\}, \qquad k = 0, \ldots, \infty, \tag{4}$$

with $A_x = A_y C$ and $b_x = b_y$. Furthermore, we assume that the system is robustly asymptotically stable

$$\lim_{n \to \infty} \max_{\|x(0)\| \leq 1, \Phi(k) \in \Omega, k=0,\ldots,n-1} \|x(n)\|_2 = 0. \tag{5}$$

We first give a definition of the concept of set invariance and then formalize the problem that is solved in this paper.

Definition 1 (Robust Positive Invariance)**.** *Given a system* (1)-(2) *satisfying* (5) *then* $\mathcal{S} \in \mathbb{R}^{n_x}$ *is a robust positive invariant set if*

$$\Phi x \in \mathcal{S}, \quad \forall x \in \mathcal{S}, \forall \Phi \in \Omega. \tag{6}$$

Remark 1. *By recursively applying* (6) *it is clear that if* $x(k) \in \mathcal{S}$ *then all future states* $x(k+i), i > 0$ *also lie within* $\mathcal{S}$*. The term* positive *means that only future states have to lie inside* $\mathcal{S}$ *(and not past states). For reasons of brevity* positive invariance *is always implicitly assumed when talking about* invariance *and moreover, hereafter, the term* invariant set *will be used to refer to* feasible robust (positive) invariant sets.[1]

Definition 2 (Feasibility)**.** *An invariant set* $\mathcal{S}$ *for a system* (1)-(2) *is* feasible *with respect to constraints* (3) *(or* (4)*) if* $\mathcal{S} \subseteq \mathcal{X}$*.*

The problem tackled in the following sections, refered to hereafter as **P1**, is the following:

Problem 1 (**P1**)**.** *Given a system* (1)-(2) *satisfying* (5) *and given state constraints* (4)*, find a feasible and robust positive invariant set* $\mathcal{S}$ *of polyhedral form*

$$\mathcal{S} = \{x \in \mathbb{R}^{n_x} | A_{\mathcal{S}} x \leq b_{\mathcal{S}}\}. \tag{7}$$

Remark 2. *Problem* ***P1*** *includes the problem of finding the set of allowable initial conditions* $x(0)$ *for which a linear system* $x(k+1) = A(k)x(k) + B(k)u(k)$ *with polytopic model uncertainty* $[A(k)\ B(k)] \in \Omega'$*, controlled by a linear state feedback controller* $u(k) = -Kx(k)$*, satisfies linear state and input constraints* $\mathcal{X} = \{x | A_x x \leq b_x\}$ *and* $\mathcal{U} = \{u | A_u u \leq b_u\}$ *for* $k = 0, \ldots, \infty$*. This can be seen by replacing* $\Phi(k), C, A_y$ *and* b_y *in* (1)- (3) *with* $A(k) + B(k)F, I_{n_x \times n_x}, [A_x^{\mathrm{T}}\ (-A_u K)^{\mathrm{T}}]^{\mathrm{T}}$ *and* $[b_x^{\mathrm{T}}\ b_u^{\mathrm{T}}]^{\mathrm{T}}$*.*

Remark 3. *The robust polyhedral invariant set of maximal size (in the sense that no points outside this set can be contained in another robust polyhedral invariant set) is unique and is called the Maximal output Admissible Set or MAS ([7]).*

3 Theoretical Extension of LTI Results

In this section an extension of the results of [7] towards LPV systems is proposed. This will be instrumental for constructing a practically implementable algorithm in the next section. Before

[1] As $x(0) \in \mathcal{S}$ implies that all corresponding future outputs will stay within the imposed constraint set $\mathcal{Y}$, an invariant set is sometimes also referred to as an *output admissible set*.

formulating a possible solution to **P1**, we summarise results proposed in [7] for LTI systems. That is, given a system (1)-(2) with $L = 1$ (i.e., an LTI system) subject to constraints (4), define $\mathcal{S}_i$ as

$$\mathcal{S}_i = \{x | A_x x \leq \mathbf{1}, A_x \Phi x \leq \mathbf{1}, \ldots, A_x \Phi^i x \leq \mathbf{1}\}. \tag{8}$$

Then, $\mathcal{S}_\infty$ is the MAS. Under certain convergence conditions, subsumed by condition (5), it can be shown:

$$\mathcal{S} \equiv \mathcal{S}_\infty = \mathcal{S}_i, \ \forall i \geq i^\star, \quad \text{with} \quad i^\star \triangleq \min\{i | S_{i+1} = S_i\} \text{ a finite integer}, \tag{9}$$

indicating that the number of constraints describing $\mathcal{S}$ is finite and that $\mathcal{S}$ is therefore polyhedral. The following algorithm then constructs the MAS $\mathcal{S}$:

Algorithm 1 (MAS for LTI systems, [7])**.** *Given a system* (1)-(2) *with* $L = 1$ *subject to constraints* (4) *and satisfying* (5)*, perform the following steps: (i) set* $i := 0$*; (ii) increase* i *until* $\mathcal{S}_{i+1} = \mathcal{S}_i$*, (iii) return* $\mathcal{S} = \mathcal{S}_i$*.*

Remark 4. *Algorithm 1 implicitly constructs forward predictions* $(x, \Phi x, \Phi^2 x, \ldots)$ *up to* i *steps ahead (for increasing* i*) and constructs the set* $\mathcal{S}_i$ *by imposing constraints* (4) *on these predictions. It is therefore called a* forward *algorithm.*

Algorithm 1 can be extended towards the LPV case by making predictions with all possible values $\Phi \in \Omega$. Due to convexity, one can capture the variability in these predictions by considering only the Φ from the vertices of Ω but nevertheless, as seen in the following, the complexity of the associated sets $\mathcal{S}_i$ still grows quickly with index 'i'.

Lemma 1. *Given* $A \in \mathbb{R}^{n_x \times n_x}, b \in \mathbb{R}^{n_x \times 1}$ *defining linear inequalites* $Ax \leq b$ *and given an uncertainty polytope* Ω *defined as in* (2)*, satisfaction of the following constraints*

$$A\Phi_1 x \leq b, \quad \ldots \quad , A\Phi_L x \leq b, \tag{10}$$

implies satisfaction of the following condition

$$A\Phi x \leq b, \quad \forall \Phi \in \Omega. \tag{11}$$

Proof. By making a weighted sum of the constraints (10) with the λ_i variables of expression (2) one obtains condition (11), which is hence satisfied if all conditions (10) are satisfied. □

Using this lemma, and by analogy with (8) which applies for $L = 1$, it is possible to redefine the sets $\mathcal{S}_i$ in order to capture all possible state predictions for LPV systems, that is, let

$$\mathcal{S}_i = \bigcap_{n=0}^{i} \overline{\mathcal{S}}_n, \quad \overline{\mathcal{S}}_n = \{x | A_{\overline{\mathcal{S}}_n} x \leq b_{\overline{\mathcal{S}}_n}\}, \tag{12}$$

where

$$A_{\overline{\mathcal{S}}_0} = A_x, \qquad b_{\overline{\mathcal{S}}_0} = b_x, \tag{13}$$

$$A_{\overline{\mathcal{S}}_n} = \begin{bmatrix} A_{\overline{\mathcal{S}}_{n-1}}\Phi_1 \\ \vdots \\ A_{\overline{\mathcal{S}}_{n-1}}\Phi_L \end{bmatrix}, \qquad b_{\overline{\mathcal{S}}_n} = \begin{bmatrix} b_{\overline{\mathcal{S}}_{n-1}} \\ \vdots \\ b_{\overline{\mathcal{S}}_{n-1}} \end{bmatrix}; \qquad n = 1, \ldots, \infty. \tag{14}$$

The following theorem now provides a potential solution to **P1**.

Theorem 1. *Define $\mathcal{S}_i$ from* (12) *and hence $\mathcal{S}_\infty$ from Algorithm 1. This $\mathcal{S}_\infty$ is a feasible robust invariant set for system* (1)-(2) *subject to constraints* (4).

Proof. By construction $\mathcal{S}_\infty \subseteq \mathcal{S}_0 \equiv \mathcal{X}$ which establishes *feasibility*. It remains to prove that $\mathcal{S}_\infty$ is *invariant*. From Lemma 1 and eqns. (14) and (12) it is clear that

$$x(k) \in \overline{\mathcal{S}}_{n+1} \quad \Rightarrow \quad x(k+1) \in \overline{\mathcal{S}}_n \;\text{ and hence }\; x(k) \in \mathcal{S}_{i+1} \quad \Rightarrow \quad x(k+1) \in \mathcal{S}_i \tag{15}$$

As $\mathcal{S}_\infty \subseteq \mathcal{S}_i,\ \forall i \geq 0$, it is then immediate that $\mathcal{S}_\infty$ is invariant. Conversely, if $x(0) \notin \mathcal{S}_\infty$, then there must exist an n such that $x(0) \notin \overline{\mathcal{S}}_n$ and therefore there must exist a sequence $\Phi(i) \in \Omega, i = 0, \ldots, k-1$ such that $x(k) \notin \mathcal{S}_0$ (i.e. $\mathcal{X}$). □

Remark 5. *Using similar arguments to earlier, it is clear that $\mathcal{S}_\infty$ is also the largest possible feasible and robust positive invariant set (that is the MAS) for the given system and constraints.*

Although $\mathcal{S}_\infty$ is a feasible positive invariant set it is not guaranteed to be polyhedral, and is therefore not necessarily a useful solution to **P1**, unless algorithm 1 terminates in finite steps.

The following two theorems show that under certain conditions the result $\mathcal{S}$ of Algorithm 1 – with the adjusted definition (12) of $\mathcal{S}_i$ – is a *polyhedral* robust invariant set for LPV systems.

Theorem 2. *Given an LPV system* (1)-(2) *and constraints* (4)*, then Algorithm 1 using definitions* (12)-(14) *constructs a valid solution to* ***P1*** *if it terminates in a finite number of iterations.*

Proof. We already know that $\mathcal{S}_\infty$ is invariant. For finite i, $\mathcal{S}_i$ is defined by a finite (albeit possibly large) number of inequalites and hence is polyhedral. □

The following theorems now indicate what might seem an intuitive result but needs formal proof, that is, under what conditions Algorithm 1 terminates in a finite number of iterations. It is shown that if $\phi_{\max} < 1$, the set $\mathcal{S}$ can be constructed with a finite number of constraints. Furthermore, it gives a numeric upper bound on the required iterations of (12)-(14).

Theorem 3. *Consider the following definitions*

$$a = \|A_{\overline{\mathcal{S}}_0}\|_2, \tag{16a}$$

$$b_{\min} = \min_i b_{\overline{\mathcal{S}}_0}(i), \tag{16b}$$

$$\phi_{\max} = \max_i \|\Phi_i\|_2, \tag{16c}$$

$$c = \max_{x \in \overline{\mathcal{S}}_0} \|x\|_2, \tag{16d}$$

with $b_{\mathcal{S}_0}(i)$ denoting the i-th element of vector $b_{\mathcal{S}_0}$ and assuming $\phi_{\max} < 1$. Then, $\mathcal{S}_k = \mathcal{S}_\infty, \forall k > n$ with n defined as

$$n = \left\lfloor \frac{\ln b_{\min} - \ln a - \ln c}{\ln \phi_{\max}} \right\rfloor. \tag{17}$$

Proof. It is sufficient to prove that $\overline{\mathcal{S}}_0 \subseteq \overline{\mathcal{S}}_k, \forall k > n$, as from (12), this establishes $\mathcal{S}_k = \mathcal{S}_{k+1},\ k \geq n$. To this end we calculate an upper bound on $\|A_{\overline{\mathcal{S}}_k} x(0)\|$ over all possible $x(0) \in \mathcal{S} \subseteq \overline{\mathcal{S}_0}$ as a

function of k:

$$\|A_{\overline{\mathcal{S}}_k} x(0)\| \leq \max_{i_0,\ldots,i_{k-1}} \|A_{\overline{\mathcal{S}}_0} \Phi_{i_{k-1}} \ldots \Phi_{i_0} x(0)\| \tag{18a}$$

$$\leq \max_{i_0,\ldots,i_{k-1}} \left(\|A_{\overline{\mathcal{S}}_0}\| \cdot \|\Phi_{i_{k-1}}\| \cdot \ldots \cdot \|\Phi_{i_0}\| \cdot \|x(0)\|\right) \tag{18b}$$

$$\leq a\phi_{\max}^k c. \tag{18c}$$

The largest element of $A_{\overline{\mathcal{S}}_k} x(0)$ is therefore bounded from above by $a\phi_{\max}^k c$. The smallest element of $b_{\overline{\mathcal{S}}_k}$ is the same as the smallest element of $b_{\overline{\mathcal{S}}_0}$ (cfr. (14)) and is therefore equal to $b_{\min}$. Hence if k satisfies the following condition, it is guaranteed that all inequalities of $\overline{\mathcal{S}}_k$ are satisfied if $x(0) \in \overline{\mathcal{S}}_0$:

$$\phi_{\max}^k \leq \frac{b_{\min}}{a\ c}, \tag{19}$$

which is equivalent with

$$k \geq \frac{\ln b_{\min} - \ln a - \ln c}{\ln \phi_{\max}}. \tag{20}$$

The inversion of the inequality is necessary since $\phi_{\max} < 1$ and therefore $\ln \phi_{\max} < 0$. It is clear that, because $k \in \mathbb{N}$, (20) is satisfied if $k > n$, which proves the theorem. □

Corollary 1. *Theorem 3 can also be applied to LPV systems for which $\phi_{\max} > 1$ as long as they are quadratically stable.*

Proof. A system is quadratically stable if there exists a quadratic Lyapunov function $J(x) = x^{\mathrm{T}} P x, P = P^{\mathrm{T}} > 0$. In this case the transformation $x' = P^{\frac{1}{2}} x$, will result in a transformed LPV system with uncertainty polytope nodes $\Phi'_i, \ldots, \Phi'_L$ satisfying $\phi'_{\max} < 1$. □

Theorem 3 provides a means of calculating the inequalities defining $\mathcal{S}$ but this method may still become computationally intractable, even for relatively small values of n. This is because the number of constraints n_c increases exponentially as a function of n (more specifically $n_c \sim \frac{L^{n+2}-1}{L-1}$). Therefore, the next section proposes a more practical algorithm.

4 Implementable Algorithm

In this section we first reformulate **P1** into a different but equivalent problem **P2** by using a modified invariance condition, after which we then propose an efficient algorithm for solving this new problem **P2**.

4.1 Reformulated invariance condition

We first define the $^-$-operator:

$$\mathcal{S}^- = \{x | \Phi x \in \mathcal{S}, \forall \Phi \in \Omega\}. \tag{21}$$

$\mathcal{S}^-$ can be interpreted as the set of all *previous states* for which it is guaranteed that the *current state* lies inside $\mathcal{S}$. Given a polyhedral set $\mathcal{S} = \{x | A_{\mathcal{S}} x \leq b_{\mathcal{S}}\}$, the set $\mathcal{S}^- = \{x | A_{\mathcal{S}^-} x \leq b_{\mathcal{S}^-}\}$ is:

$$A_{\mathcal{S}^-} = \begin{bmatrix} A_{\mathcal{S}}\Phi_1 \\ \vdots \\ A_{\mathcal{S}}\Phi_L \end{bmatrix}, \qquad b_{\mathcal{S}^-} = \begin{bmatrix} b_{\mathcal{S}} \\ \vdots \\ b_{\mathcal{S}} \end{bmatrix}. \tag{22}$$

This now enables the formulation of a necessary and sufficient condition for robust positive invariance.

Lemma 2. *A set $\mathcal{S}$ is a robust positive invariant set for the system* (1) *iff*

$$\mathcal{S} \subseteq \mathcal{S}^-. \tag{23}$$

Proof. Satisfaction of (23) gives $x \in \mathcal{S} \;\Rightarrow\; x \in \mathcal{S}^-$ and therefore – by applying Lemma 1 – also $\Phi x \in \mathcal{S}$, which proves that (23) is a *sufficient* condition for robust positive invariance. On the other hand, if there exists a state $x \in (\mathcal{S} \setminus \mathcal{S}^-)$ then there exists $\Phi \in \Omega$ such that $\Phi x \notin \mathcal{S}$, which proves that (23) is also a *necessary* condition. □

Lemma 2 is a generalization of condition $\mathcal{S}_i = \mathcal{S}_{i+1}$, used here to facilitate the generalization for the LPV case of the LTI invariance condition [7, Theorem 2.2]. This can be seen by observing that $\mathcal{S}_{i+1} \equiv \mathcal{S}_i \cap \overline{\mathcal{S}}_{i+1} = \mathcal{S}_i \cap \overline{\mathcal{S}}_i^-$, which allows $\mathcal{S}_i = \mathcal{S}_{i+1}$ to be rewritten as $\mathcal{S}_n \subseteq \overline{\mathcal{S}}_i^-$ and hence

as $\mathcal{S}_i \subseteq \mathcal{S}_i^-$, which is clearly a special case of condition (23). Hence, Lemma 2 enables us to reformulate problem **P1** into the following equivalent problem, refered to hereafter as **P2**.

Problem 2 (P2). *Given a system* (1)-(2) *satisfying* (5) *and given the constraints* (3)*, find matrices* $A_{\mathcal{S}}$ *and* $b_{\mathcal{S}}$ *such that the set* $\mathcal{S} = \{x \in \mathbb{R}^{n_x} | A_{\mathcal{S}} x \leq b_{\mathcal{S}}\}$ *satisfies*

$$\mathcal{S} \subseteq \mathcal{S}^- \equiv \{x \in \mathbb{R}^{n_x} | A_{\mathcal{S}}^- x \leq b_{\mathcal{S}}^-\}, \tag{24a}$$

$$\mathcal{S} \subseteq \mathcal{X}, \tag{24b}$$

with $A_{\mathcal{S}}^- \equiv [A_{\mathcal{S}}\Phi_1; \ldots; A_{\mathcal{S}}\Phi_L]$ *and* $b_{\mathcal{S}}^- \equiv [b_{\mathcal{S}}; \ldots; b_{\mathcal{S}}]$.

4.2 Algorithm for solving P2

This section formulates an algorithm for solving **P2** that starts with the set $\mathcal{S}_0 = \mathcal{X}$ and then iteratively adds constraints from $\mathcal{S}_1, \mathcal{S}_2, \ldots$ in order to satisfy (23). Elements of matrices are denoted using Matlab notation; e.g. $A_{(i,:)}$ denotes the i-th row of A.

Algorithm 2. *Given a linear system* (1)-(2) *satisfying* (5) *and given the constraints* (4)*, perform the following steps :*

1. *Set the initial values* $A_{\mathcal{S}} := A_x, \;\; b_{\mathcal{S}} := b_x$. *as defined in* (4) *and initialize the index* $i := 1$.

2. *Iterate while* i *is not strictly larger than the number of rows in* $A_{\mathcal{S}}$*:*

   (a) *Select row* i *from* $A_{\mathcal{S}}$ *and* $b_{\mathcal{S}}$*:* $\quad a^{\mathrm{T}} = (A_{\mathcal{S}})_{(i,:)}, \; b = (b_{\mathcal{S}})_i$.

   (b) *Check whether adding constraints* $a^{\mathrm{T}}\Phi_j x \leq b, \; j = 1, \ldots, L$ *to* $A_{\mathcal{S}}, b_{\mathcal{S}}$ *decreases the size of* $\mathcal{S}$*, by solving the following LP for* $j = 1, \ldots, L$*:*

$$c_j = \max_x \quad a^{\mathrm{T}}\Phi_j x - b, \quad \text{s.t.} \quad A_{\mathcal{S}} x \leq b_{\mathcal{S}}. \tag{25}$$

   For each $j = 1, \ldots, L$*, if* $c_j > 0$*, then add the constraint* $a^{\mathrm{T}}\Phi_j x \leq b$ *to* $A_{\mathcal{S}}, b_{\mathcal{S}}$*:*

$$A_{\mathcal{S}} := \begin{bmatrix} A_{\mathcal{S}} \\ a^{\mathrm{T}}\Phi_j \end{bmatrix}, \qquad\qquad b_{\mathcal{S}} := \begin{bmatrix} b_{\mathcal{S}} \\ b \end{bmatrix}. \tag{26}$$

   Set $\mathcal{T}_i = \{x | A_{\mathcal{S}} x \leq b_{\mathcal{S}}\}$.

(c) Increment $i := i + 1$.

3. Return $\mathcal{S} = \{x | A_{\mathcal{S}} x \leq b_{\mathcal{S}}\}$.

The next two lemmata show that the sets in Algorithm 2 are strictly nested with each iteration and this enables a proof of correctness. A theorem then uses these to establish convergence for Algorithm 2.

Lemma 3. *The intermediate sets* $\mathcal{T}_i = \{x | A_{\mathcal{S}} x \leq b_{\mathcal{S}}\}, i = 1, \ldots, n_c$ *taking* $A_{\mathcal{S}}$, $b_{\mathcal{S}}$ *from step 2b of Algorithm 2, with* n_c *denoting the number of constraints defining* $\mathcal{S}$*, always satisfy the condition* $\mathcal{S} \subseteq \mathcal{T}_i$.

Proof. Because constraints are only added to $A_{\mathcal{S}}, b_{\mathcal{S}}$ in each iteration, it is guaranteed by construction that $\mathcal{T}_{i+1} \subseteq \mathcal{T}_i$. Since $\mathcal{S} \equiv \mathcal{T}_{n_c}$ this then proves the lemma. □

Lemma 4 (Correctness)**.** *If Algorithm 2 terminates in a finite number of iterations then the resulting set* $\mathcal{S} = \{x | A_{\mathcal{S}} x \leq b_{\mathcal{S}}\}$ *is a valid solution to* ***P2****.*

Proof. From the initialization step 1) ($\mathcal{T}_0 = \mathcal{X}$) and lemma 3, is is clear that $\mathcal{S}$ will satisfy (24b). Satisfaction of (24a) also follows from Lemma 3 as, on termination of the algorithm, the variable i is equal to the number of rows in $A_{\mathcal{S}}$. □

Theorem 4 (Convergence)**.** *Under the same conditions as Theorem 3, Algorithm 2 will terminate in a finite number of iterations.*

Proof. By construction Algorithm 2 only uses constraints from the sets $\overline{\mathcal{S}}_i, i = 1, \ldots, \infty$ (see (14)) to add to $\mathcal{S}$. Moreover, Theorem 3 showed that all constraints of sets $\overline{\mathcal{S}}_i, i > n$, with n given by (17), are redundant with respect to $\mathcal{S}_0$ and therefore also with respect to $\mathcal{S} \subseteq \mathcal{S}_0$. Hence, Algorithm 2 must reach the termination condition of step 3) in a finite number of iterations, (beyond which one is implicity adding inequalities from $\overline{\mathcal{S}}_i, i > n$), which proves the lemma. □

4.3 Garbage collection

Algorithm 2 gives significant reductions in complexity as many of the (theoretically possible) exponential number of constraints making up $\mathcal{S}$ are redundant, and step 2b) adds only non-redundant constraints (w.r.t. $\mathcal{T}_i$). Illustrations of this are given in Section 6. Furthermore, as $\mathcal{T}_{i+1} \subseteq \mathcal{T}_i,\ \forall i$, one can reduce complexity further still by regularly rechecking the redundancy of constraints added in previous iterations. Removing redundant constraints, here denoted *Garbage Collection*, is a standard procedure, so the following summarises only those points pertinent to this paper. *Garbage Collection* is deployed as an additional step 2d to algorithm 2 but it is not executed every iteration. For instance it may be activated only if n_c has grown by a given factor (say 50%) or increment.

Algorithm 3 (Garbage Collection)**.** *Given a set $\mathcal{S} = \{x | A_{\mathcal{S}} x \leq b_{\mathcal{S}}\}$, with $A_{\mathcal{S}} \in \mathbb{R}^{n_c \times n_x}$, execute the following for $j = n_c, \ldots, 1$:*

(a) Check for the redundancy of the jth row and if the jth row is redundant:

1. *Remove the jth row, hence reducing by one the number of inequalities.*
2. *If $i > j$ set $i := i - 1$*

(b) Return the updated matrices $A_{\mathcal{S}}, b_{\mathcal{S}}$ and updated i.

Remark 6. *It should be emphasised that the possibility of performing garbage collection during the construction of the invariant set is essential to practical computations and is possible only due to novel adoption of invariance condition* (23)*; notably this also allows the iteration to commence from any $\mathcal{S}_0$ satisfying $\mathcal{S} \subseteq \mathcal{S}_0 \subseteq \mathcal{X}$. Furthermore, the addition of garbage collection to Algorithm 2 does not invalidate the arguments used in the proofs of Lemmas 3, 4 and Theorem 4 since $\mathcal{T}_i$, and therefore $\mathcal{S}$, are not modified by the removal of the redundant constraints.*

4.4 Extensions

For completeness, this subsection discusses briefly two extensions that allow us to take account of bounded disturbances and contraction constraints into account. Both methods use a modified definition of the $^-$-operator.

4.4.1 Bounded disturbances

Consider an LPV system with bounded disturbances:

$$x(k+1) = \Phi(k)x(k) + w(k), \tag{27}$$

with $w(k) \in \mathcal{W} = \text{Co}\{w_1, \ldots, w_N\}, \forall k$ and $\Phi(k) \in \Omega, \forall k$. Algorithm 2 can be deployed if the definition of the $^-$-operator is modified to:

$$\mathcal{S}^- = \{x | \Phi x + w \in \mathcal{S}, \forall \Phi \in \Omega, \forall w \in \mathcal{W}\}. \tag{28}$$

For instance, given a set $\mathcal{S} = \{x | A_{\mathcal{S}} x \leq b_{\mathcal{S}}\}$, the set $\mathcal{S}^- = \{x | A_{\mathcal{S}^-} x \leq b_{\mathcal{S}^-}\}$ is:

$$A_{\mathcal{S}^-} = \begin{bmatrix} A_{\mathcal{S}}\Phi_1 \\ A_{\mathcal{S}}\Phi_1 \\ \vdots \\ A_{\mathcal{S}}\Phi_L \\ A_{\mathcal{S}}\Phi_L \end{bmatrix}, \qquad b_{\mathcal{S}^-} = \begin{bmatrix} b_{\mathcal{S}} - A_{\mathcal{S}}w_1 \\ b_{\mathcal{S}} - A_{\mathcal{S}}w_2 \\ \vdots \\ b_{\mathcal{S}} - A_{\mathcal{S}}w_{N-1} \\ b_{\mathcal{S}} - A_{\mathcal{S}}w_N \end{bmatrix}. \tag{29}$$

Convergence of Algorithm 2 in this setting can be proven if $\max_i \|w_i\|$ is sufficiently small.

4.4.2 Contraction constraints

For various reasons it might be favorable to include contraction constraints when constructing the invariant set, meaning that one requires an invariant set $\mathcal{S}$ with the property:

$$\Phi x \in c\mathcal{S}, \qquad \forall \Phi \in \Omega, \forall x \in \mathcal{S}, \tag{30}$$

where $c \in (0, 1)$ and $c\mathcal{S} = \{x | A_{\mathcal{S}} x \leq c b_{\mathcal{S}}\}$. This can be obtained with Algorithm 2 by modifying $\mathcal{S}^- = \{x | A_{\mathcal{S}^-} x \leq b_{\mathcal{S}^-}\}$:

$$A_{\mathcal{S}^-} = \begin{bmatrix} A_{\mathcal{S}}\Phi_1 \\ \vdots \\ A_{\mathcal{S}}\Phi_L \end{bmatrix}, \qquad b_{\mathcal{S}^-} = c \cdot \begin{bmatrix} b_{\mathcal{S}} \\ \vdots \\ b_{\mathcal{S}} \end{bmatrix}, \tag{31}$$

which comes down to computing a regular invariant set for an LPV system with uncertainty polytope nodes $\frac{\Phi_i}{c}$. Correctness is trivial; convergence can be proven if $\max_j \|\Phi_j\| < c$.

5 Robust MPC using Polyhedral Invariant Sets

Model based Predictive Control is a control paradigm that, given a system description

$$x(k+1) = A(k)x(k) + B(k)u(k), \qquad k = 0, \ldots, \infty, \tag{32}$$

$$[A(k)\ B(k)] \in \Omega' \equiv \mathrm{Co}\{[A_1\ B_1], \ldots, [A_L\ B_L]\}, \qquad k = 0, \ldots, \infty, \tag{33}$$

and given state constraints (4) and input constraints

$$u(k) \in \mathcal{U} \equiv \{u | A_u u \leq b_u\}, \qquad k = 0, \ldots, \infty, \tag{34}$$

solves at each time instant k a finite-horizon open-loop dynamic optimization problem after which only the first input of the resulting optimal input sequence is applied to the system. In the LTI case ($L = 1$) the online optimization problem typically reduces to a QP, while in the LPV case ($L > 1$) a minimization of a worst-case (over all possible system dynamics) objective function is performed. An overview can be found in [5, 12, 17].

Stability and recursive feasibility can be obtained by: 1) considering a terminal controller that is implicitly applied beyond the prediction horizon; 2) constraining the terminal state (the last state within the prediction horizon) to lie within a feasible invariant set for the closed loop system formed with the terminal controller and 3) an additional cost term that overbounds the worst-case control cost beyond the prediction horizon. For an overview of this stability framework we refer to [14].

While nominal MPC algorithms [18, 20] typically make use of polyhedral invariant sets, robust MPC algorithms usually make use of ellipsoidal invariant sets [3, 10, 21]. However, the latter have the disadvantage of resulting in a smaller feasible region, dealing poorly with non-symmetrical constraints and leading to complex LMI-based optimization problems. Significantly, this section

proposes a new robust MPC algorithm based on QP. It uses the polyhedral invariant sets of the earlier sections and can be seen as an extension of the algorithm introduced in [10] towards polyhedral invariant sets or as an extension of [18] towards LPV systems.

In the first subsection the new algorithm is formulated as a min-max optimization problem. In subsequent subsections synthesis results are given for the construction of the optimization constraints and objective function. Finally, the algorithm is proven to be recursively feasible and robustly stabilizing.

5.1 MPC Algorithm formulation

The algorithm introduced in this subsection is closely related to the algorithms introduced in [10, 18] and is a slight generalization of the algorithm introduced in [16]. The algorithm uses a finite prediction horizon of length N and parameterizes the system inputs as follows:

$$u(k+i|k) = -Kx(k+i|k) + c(k+i|k), \qquad i = 0, \ldots, N-1, \tag{35a}$$

$$u(k+i|k) = -Kx(k+i|k), \qquad i = N, \ldots, \infty, \tag{35b}$$

where $u(k+i|k) \in \mathbb{R}^{n_u \times 1}$ and $x(k+i|k) \in \mathbb{R}^{n_x \times 1}$ respectively denote the system input and state vectors at time $k+i$, as predicted at time k, with n_x, n_u the state and input dimensions. $K \in \mathbb{R}^{n_u \times n_x}$ is a state feedback matrix that robustly asymptotically stabilizes system (32)-(33), while $c(k+i|k) \in \mathbb{R}^{n_u \times 1}, i = 0, \ldots, N-1$ are free control moves that are superimposed on the input signal provided by the feedback gain K.

Constraint handling within the horizon is done explicitly

$$\left.\begin{array}{l} x(k+i|k) \in \mathcal{X} \\ u(k+i|k) \in \mathcal{U} \end{array}\right\}, \quad \forall [A(j)\ B(j)] \in \Omega', j = 0, \ldots, i-1, \quad i = 0, \ldots, N-1, \tag{36}$$

while constraint handling beyond the horizon is done implicitly by constraining the terminal state to lie within a set $\mathcal{S}_N$

$$x(k+N|k) \in \mathcal{S}_N, \quad \forall [A(j)\ B(j)] \in \Omega', j = 0, \ldots, N-1, \tag{37}$$

with the set $\mathcal{S}_N$ a feasible invariant set for the closed loop system formed by the terminal controller and system (32)-(33), which can be written in the form (1)-(2), with

$$\Phi_i = A_i - B_i K, \qquad i = 1, \ldots, L, \tag{38}$$

subject to constraints

$$x(k) \in \{x | A_x x \leq b_x, -A_u K x \leq b_u\}, \qquad k = 0, \ldots, \infty. \tag{39}$$

The aim is to use an objective function $J(x(k), \mathbf{c}(k))$ that is an upper bound to the infinite-horizon worst case cost over all possible dynamics

$$J(x(k), \mathbf{c}(k)) \geq \max_{[A(j)\ B(j)] \in \Omega', j=0,\ldots,\infty} \sum_{i=0}^{\infty} \left(x(k+i|k)^{\mathrm{T}} Q x(k+i|k) + u(k+i|k)^{\mathrm{T}} R u(k+i|k) \right). \tag{40}$$

$Q = Q^{\mathrm{T}} \in \mathbb{R}^{n_x \times n_x}$ and $R = R^{\mathrm{T}} \in \mathbb{R}^{n_u \times n_u}$ denote positive definite state and input weighting matrices, while $\mathbf{c}_N(k) = [c(k|k)^{\mathrm{T}}, \ldots, c(k+N-1|k)^{\mathrm{T}}]^{\mathrm{T}}$ denotes the sequence of free control moves considered at time k. The following algorithm now brings together the above elements :

Algorithm 4. *Given a system* (32)-(33)*, constraints* (4),(34)*, cost weighting matrices* $Q = Q^{\mathrm{T}} > 0$ *and* $R = R^{\mathrm{T}} > 0$ *and an objective function* $J(x(k), \mathbf{c}_N(k))$ *satisfying* (40)*, solve at each time instant* k*, given the current state* $x(k)$ *the following optimization problem :*

$$\min_{\mathbf{c}_N(k)} \quad J(x(k), \mathbf{c}(k)), \tag{41}$$

subject to (35),(36),(37) *and apply the input* $u(k) = -Kx(k) + c^{\mathrm{o}}(k|k)$ *to the system, with* $c^{\mathrm{o}}(k|k)$ *denoting the optimal value of* $c(k|k)$ *resulting from the above optimization problem.*

In the following two subsections efficient methods are formulated for constructing the constraint sets (35),(36),(37) and the objective function (40).

5.2 Constraint synthesis

The constraints of Algorithm 4 can be calculated by applying Algorithm 2 to the augmented LPV system $\tilde{x}(k+1) = \tilde{\Phi}(k)\tilde{x}(k)$ with the nodes of the uncertainty polytope defined as :

$$\tilde{\Phi}_i = \begin{bmatrix} A_i - B_i K & B_i & 0 \\ 0 & 0 & I_{((N-1)\cdot n_u,(N-1)\cdot n_u)} \\ 0 & 0 & 0 \end{bmatrix}, \qquad i = 1, \ldots, L, \tag{42}$$

subject to constraints $A_{\tilde{x}}\tilde{x}(k) \leq b_{\tilde{x}}, k = 0, \ldots, \infty$ with $A_{\tilde{x}}$ defined as

$$A_{\tilde{x}} = \begin{bmatrix} A_x & 0 & 0 \\ -A_u K & A_u & 0 \end{bmatrix}, \qquad B_{\tilde{x}} = \begin{bmatrix} b_x \\ b_u \end{bmatrix}, \tag{43}$$

resulting in the invariant set $\mathcal{S}_{\text{aug}}$.

The following lemmata now show the link between the constructed invariant set $\mathcal{S}_{\text{aug}}$ and constraints (35), (36),(37). Specifically, lemmata 5-7 show that the constraint $[x(k)^{\mathrm{T}}\ \mathbf{c}_N(k)^{\mathrm{T}}]^{\mathrm{T}} \in \mathcal{S}_{\text{aug}}$ is equivalent with constraints (35),(36), (37) of Algorithm 4. However, first it is necessary to validate that the augmented system (42) captures the dynamics of the dynamic optimization problem of Algorithm 4 using the input parametrization (35) and $\tilde{x}(k) := [x(k)^{\mathrm{T}}\ \mathbf{c}_N(k)^{\mathrm{T}}]^{\mathrm{T}}$ as an initial state.

Lemma 5. *Given a value $x(k)$, set $\tilde{x}(k) := [x(k)^{\mathrm{T}}\ \mathbf{c}_N(k)^{\mathrm{T}}]^{\mathrm{T}}$. The first n_x components of $\tilde{x}(k+i), i = 1, \ldots, \infty$ now correspond to the state predictions $x(k+i|k), i = 1, \ldots, N$ of Algorithm 4 using inputs* (35).

The proof is obvious and hence omitted.

Lemma 6. *If $[x(k)^{\mathrm{T}}\ \mathbf{c}_N(k)^{\mathrm{T}}]^{\mathrm{T}} \in \mathcal{S}_{\text{aug}}$, then constraints* (36) *are satisfied.*

Proof. Due to the robust invariance property of $\mathcal{S}_{\text{aug}}$, we know $\tilde{x}(k+i) \in \mathcal{S}_{\text{aug}}, i = 1, \ldots, \infty, \forall [A(k+i)\ B(k+i)] \in \Omega', i = 0, \ldots, N-1$. Then Lemma 5 and the equivalence of (43) to (36) with control law (35) proves the lemma. □

Lemma 7. *If $[x(k)^{\mathrm{T}}\ \mathbf{c}_N(k)^{\mathrm{T}}]^{\mathrm{T}} \in \mathcal{S}_{\text{aug}}$, then constraint* (37) *is satisfied.*

Proof. Lemma 5 gives that $\tilde{x}(k+N) = [x(k+N|k)^{\mathrm{T}}\ 0^{\mathrm{T}}]^{\mathrm{T}}, \forall [A(k+i)\ B(k+i)] \in \Omega', i = 0, \ldots, N-1$ and hence through the robust invariance property of $\mathcal{S}_{\mathrm{aug}}$ and the structure of system (42) it is consequently guaranteed that $[x(k+N+i|k)^{\mathrm{T}}\ 0^{\mathrm{T}}]^{\mathrm{T}} \in \mathcal{S}_{\mathrm{aug}},\ \forall i \geq 0$ which in turn implies constraint satisfaction, from the equivalence of (43) to (37), for all future time instants. Finally, robust constraint satisfaction for $x(k+N+i|k),\ \forall i \geq 0$ with feedback (35b) implies that $x(k+N|k) \in \mathcal{S}_N$. □

5.3 Cost function synthesis

As with earlier works, this paper adopts an objective function which is an upper bound on the worst case finite-horizon control cost. The worst case cost function (40) can be represented as

$$J(x(k), \mathbf{c}_N(k)) = \tilde{x}(k)^{\mathrm{T}} P \tilde{x}(k), \tag{44}$$

with $P = P^{\mathrm{T}} \in \mathbb{R}^{(n_x+(N\cdot n_u))\times(n_x+(N\cdot n_u))}$ satisfying

$$P - \tilde{\Phi}_i^{\mathrm{T}} P \tilde{\Phi}_i > \Gamma_x^{\mathrm{T}} Q \Gamma_x + \Gamma_u^{\mathrm{T}} R \Gamma_u, \qquad i = 1, \ldots, L, \tag{45}$$

with $\Gamma_x = [I_{(n_x,n_x)}\ 0]$ and $\Gamma_u = [-K\ I_{(n_u,n_u)}\ 0]$ and $\tilde{\Phi}_i, i = 1, \ldots, L$ defined as in (42). P can efficiently be calculated using the convex optimization problem

$$\min_{P=P^{\mathrm{T}}>0} \ \mathrm{tr}(P), \qquad \text{subject to (45)}. \tag{46}$$

Lemma 8. *The cost function $J(x(k), \mathbf{c}_N(k))$ as defined in* (44)-(45) *satisfies* (40).

Proof. Eqn. (45) guarantees that if $\tilde{x}(k) = [x(k)^{\mathrm{T}}\ \mathbf{c}_N(k)^{\mathrm{T}}]^{\mathrm{T}}$ then

$$\tilde{x}(k+j)^{\mathrm{T}} P \tilde{x}(k+j) - \tilde{x}(k+j)^{\mathrm{T}} \tilde{\Phi}(k+j)^{\mathrm{T}} P \tilde{\Phi}(k+j) \tilde{x}(k+j) \geq$$
$$x(k+j|k)^{\mathrm{T}} Q x(k+j|k) + u(k+j|k)^{\mathrm{T}} R u(k+j|k), \tag{47}$$

for any $\tilde{x}(k+j) = [x(k+j|k)^{\mathrm{T}}\ c(k+j|k)^{\mathrm{T}}\ \ldots\ c(k+N-1|k)^{\mathrm{T}}\ 0]$, any $j > 0$ and any $[A(k+j)\ B(k+j)] \in \Omega'$, which, after summation over $j = 0, \ldots, \infty$, proves the lemma. □

Remark 7. *In case the feedback gain K is optimal with respect to some criterion (e.g., H_2, H_∞) it might be preferable [18] to minimize $\sum_{i=0}^{N-1} c(k+i|k)^{\mathrm{T}} c(k+i|k)$ in order to minimize the deviation from the unconstrained optimal. In that case one can choose $P = diag(0, I, \ldots, I)$.*

5.4 Recursive feasibility and robust stability

The previous subsections can now be summarized into the following algorithm that – as opposed to Algorithm 4 – is practically implementable. Notably, due to the fact that $J(x(k), \mathbf{c}_N(k))$ is a quadratic function and $\mathcal{S}_{\mathrm{aug}}$ is a polyhedral set, the optimization problem (48) is a QP.

Algorithm 5. *Given a system* (32)-(33)*, constraints* (4),(34)*, cost weighting matrices $Q = Q^{\mathrm{T}} > 0$ and $R = R^{\mathrm{T}} > 0$, a polyhedral feasible invariant set $\mathcal{S}_{\mathrm{aug}}$ for* (42)-(43) *and an objective function $J(x(k), \mathbf{c}_N(k)) = [x(k)^{\mathrm{T}}\ \mathbf{c}_N(k)^{\mathrm{T}}] P [x(k)^{\mathrm{T}}\ \mathbf{c}_N(k)^{\mathrm{T}}]^{\mathrm{T}}$ calculated using* (46)*, solve at each time k, given the current state $x(k)$ the following optimization problem*

$$\min_{\mathbf{c}_N(k)} \ J(x(k), \mathbf{c}_N(k)), \qquad \text{subject to} \qquad [x(k)^{\mathrm{T}}\ \mathbf{c}_N(k)^{\mathrm{T}}]^{\mathrm{T}} \in \mathcal{S}_{\mathrm{aug}}, \tag{48}$$

and apply $u(k) = -Kx(k) + c^{\mathrm{o}}(k|k)$ to the plant with $c^{\mathrm{o}}(k|k)$ the optimal value of $c(k|k)$ resulting from (48)*.*

Lemma 9 (Recursive Feasibility)**.** *Algorithm 5, when applied in a receding horizon fashion, is feasible at time $k > 0$ if it is feasible at time $k = 0$.*

Proof. Given optimal values $\mathbf{c}_N^{\mathrm{o}}(k) = [c^{\mathrm{o}}(k|k)^{\mathrm{T}}\ \ldots\ c^{\mathrm{o}}(k+N-1|k)^{\mathrm{T}}]^{\mathrm{T}}$ to the feasible optimization problem (48) at time k it is possible to construct a feasible set of control moves $\mathbf{c}_N^{\mathrm{f}}(k+1)$ at time $k+1$ as $\mathbf{c}_N^{\mathrm{f}}(k+1) = [c^{\mathrm{o}}(k+1|k)^{\mathrm{T}}\ \ldots\ c^{\mathrm{o}}(k+N-1|k)^{\mathrm{T}}\ 0]^{\mathrm{T}}$. Based on the invariance property of $\mathcal{S}_{\mathrm{aug}}$ it is now guaranteed that $[x(k+1)^{\mathrm{T}}\ \mathbf{c}_N^{\mathrm{f}}(k+1)^{\mathrm{T}}]^{\mathrm{T}} \in \mathcal{S}_{\mathrm{aug}}, \forall [A(k)\ B(k)] \in \Omega'$, which shows that the optimization problem (48) is also feasible at time $k+1$. Recursive application of this observation now proves the theorem. □

Theorem 5 (Robust Stability and Convergence)**.** *Algorithm 5, if feasible at time $k = 0$, robustly asymptotically stabilizes system* (32)-(33) *subject to constraints* (4),(34)*.*

Proof. Robust asymptotic stability is established by first ensuring feasibility (lemma 9) and then proving that $J^{\mathrm{o}}(k) \triangleq J(x(k), \mathbf{c}_N^{\mathrm{o}}(k))$ is monotonically decreasing as a function of k and hence acts as a Lyapunov function. Given optimal values $\mathbf{c}_N^{\mathrm{o}}(k)$ at time k, we use the feasible values $\mathbf{c}_N^{\mathrm{f}}(k+1)$ as constructed in the proof of lemma 9 to calculate $J^{\mathrm{f}}(k+1) \triangleq J(x(k+1), \mathbf{c}_N^{\mathrm{f}}(k+1))$. Due to the fact that $[x(k+1)^{\mathrm{T}}\ \mathbf{c}_N^{\mathrm{f}}(k+1)^{\mathrm{T}}]^{\mathrm{T}} = \tilde{\Phi}(k)[x(k)^{\mathrm{T}}\ \mathbf{c}_N^{\mathrm{o}}(k)^{\mathrm{T}}]^{\mathrm{T}}$ and that P satisfies (45), it is guaranteed that $J^{\mathrm{f}}(k+1) < J^{\mathrm{o}}(k), \forall [A(k)\ B(k)] \in \Omega'$. Therefore it is also guaranteed that $J^{\mathrm{o}}(k+1) < J^{\mathrm{o}}(k), \forall [A(k)\ B(k)] \in \Omega'$, which establishes robust asymptotic stability. Convergence follows as $J \to 0 \ \Rightarrow\ x \to 0$. □

6 Example

In this section a numerical example is discussed that illustrates the algorithms proposed in previous sections. We consider a system with $n_x = 2, n_u = 1, L = 2$, described by matrices

$$A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix}, \tag{49a}$$

$$B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \tag{49b}$$

subject to constraints (for $k = 0, \ldots, \infty$)

$$-1 \leq u(k) \leq 0.5, \tag{50a}$$

$$\begin{bmatrix} -10 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 10 \\ 10 \end{bmatrix}. \tag{50b}$$

An asymptotically stabilizing feedback gain $K = [0.5\ 0.3]$ is chosen, while cost weighting matrices $Q = I, R = 0.01$ are used for the control objective.

6.1 Polyhedral invariant sets

First we consider the volume of the polyhedral invariant sets, more specifically the MAS, arising from Algorithm 2. These are computed for closed-loop systems based on system (49) with two alternative controllers (figs. 1, 2) and tighter state constraints (fig. 3). In addition, the MAS of

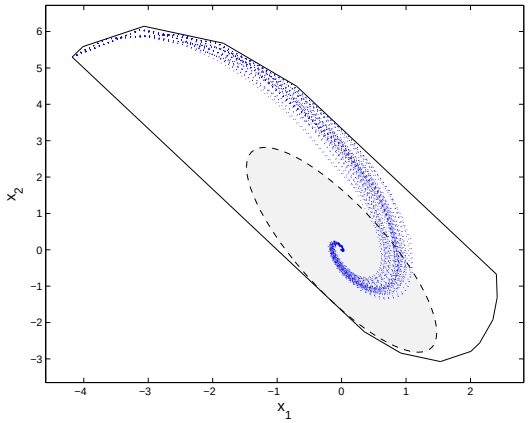

Figure 1: Ellipsoidal and polyhedral invariant set for the closed loop system formed by (49)-(50) and feedback law $u = [-0.5\ 0.3]x$. 50 trajectories for different time-varying dynamics $[A(k)\ B(k)], k = 0, \ldots, \infty$ are plotted to illustrate the invariance property.

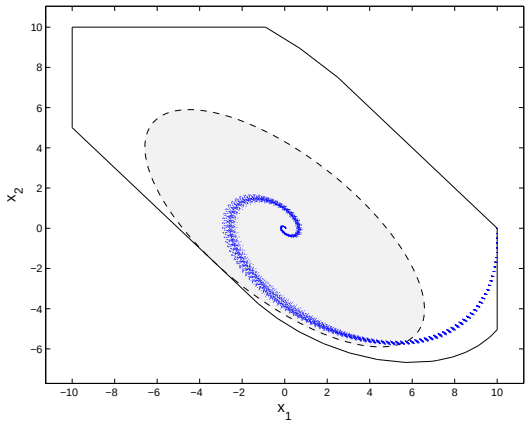

Figure 2: Ellipsoidal and polyhedral invariant set for the closed loop system formed by (49)-(50) and feedback law $u = [-0.1\ 0.1]x$. 50 trajectories for different time-varying dynamics $[A(k)\ B(k)], k = 0, \ldots, \infty$ are plotted to illustrate the invariance property.

Algorithm 2 are contrasted with ellipsoidal invariant sets (using results from [9]). The polyhedral invariant sets are significantly larger in each case and moreover can cope more effectively with the non-symmetrical constraints. Several closed loop trajectories are also plotted to show that the sets are indeed robust invariant.

Secondly, some observations on the vast reductions in complexity are pertinent. Figure 4 shows the tree structure of the constraints defining the MAS of Figure 1. The maximum prediction depth required is $n = 5$. Combining this fact with the 6 constraints of (50) and $L = 2$, the number of constraints that would be obtained when using Algorithm 1 is $6\frac{2^{5+2}-1}{2-1} = 762$. Out of these 762

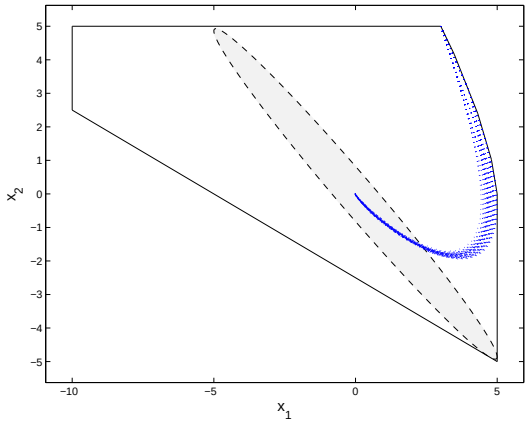

Figure 3: Ellipsoidal and polyhedral invariant set for the closed loop system formed by (49) and feedback law $u = [-0.1\ 0.1]x$. The system was subjected to constraints $[-10\ -10]^{\mathrm{T}} \leq x(k) \leq [5\ 5]^{\mathrm{T}}$ and $-2 \leq u(k) \leq 0.5,\ \forall k$. 50 trajectories for different time-varying dynamics $[A(k)\ B(k)], k = 0, \ldots, \infty$ are plotted to illustrate the invariance property.

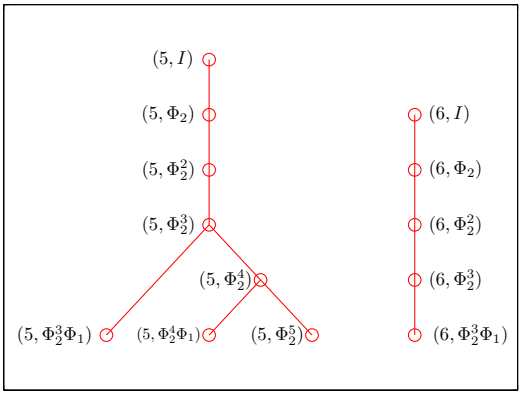

Figure 4: Tree structure of the constraints defining the invariant set depicted in Fig. 1. The notation (i, M) denotes the constraint $a_i^{\mathrm{T}} M x \leq b_i$, with a_i^{T}, b_i denoting the i-th rows of $A_{\mathcal{Y}}$ and $b_{\mathcal{Y}}$ respectively.

constraints, Algorithm 2 only retains 13 non-redundant constraints and terminates in just 1.45 seconds (1.6GHz PC with 512MB RAM using MATLAB 6.5). Similar observations can be made for the other invariant sets depicted.

6.2 Robust MPC

This subsection compares the feasibility and closed-loop performance of Algorithm 5 to the algorithm introduced in [10] (but modified to use the same objective function (44) calculated using (46)). Hence the only difference between both methods is the use of polyhedral versus ellipsoidal

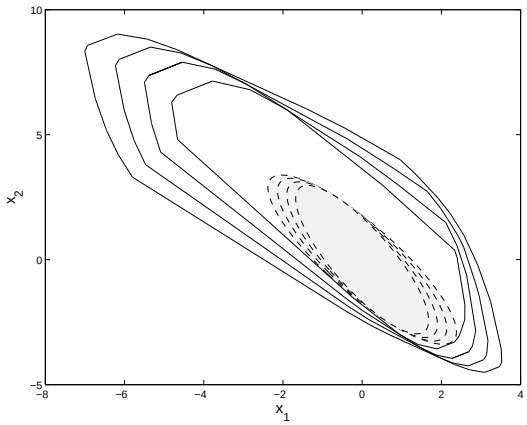

Figure 5: Feasible regions for Algorithm 5 (solid) and the algorithm from [10] (dashed) for system (49)-(50), feedback law $u = [-0.5 \ -0.3]x$ and prediction horizons $N = 1, \ldots, 4$.

invariant sets. Invariant sets for prediction horizon $N = 0$ are depicted in Figure 1. The feasible regions for $N = 1, \ldots, 4$ are depicted in Figure 5 from which it is clear that the feasible regions for Algorithm 5 are significantly larger compared to the ellipsoidal feasible regions.

Closed loop trajectories for both algorithms are depicted in Figure 6. The trajectories of Algorithm 5 are markedly different to those using ellipsoidal sets and in fact can be observed in (e.g. figure 7a,b) to converge to the origin more quickly. Furthermore, the trajectories of Algorithm 5 are non-symmetrical with respect to the origin, indicating that the algorithm can take advantage of the non-symmetrical input constraints. Figure 7 gives more detail on the symmetry issue by displaying the state and input trajectories for both algorithms for initial states $x(0) = [1.75 \ 0]^{\mathrm{T}}$ and $x(0) = [-1.75 \ 0]^{\mathrm{T}}$. The trajectories for the algorithm from [10] are clearly symmetrical, whereas the trajectories of Algorithm 5 take advantage of the non-symmetrical input constraints and are non-symmetrical. Hence, whereas the total simulation cost for the former algorithm is 7.3285 for both initial states, the simulation cost for Algorithm 5 is lower and also different for both initial states : 4.2376 and 6.7603 respectively. Further evidence is seen in figure 7c, where Algorithm 5 uses the full control authority for both initial conditions whereas, in one case, the algorithm from [10] never reaches the input constraints.

Of course Algorithm 5 is also valid, and hence has better performance, in a large region outside

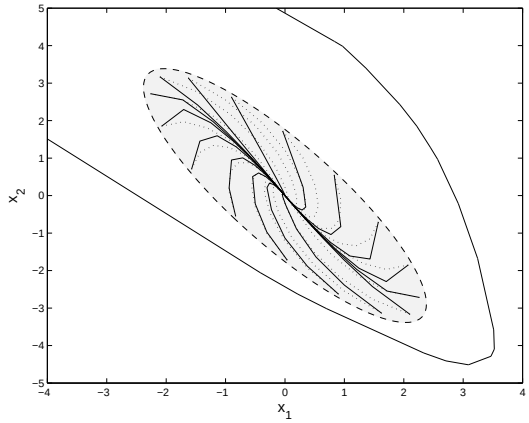

Figure 6: Trajectories for Algorithm 5 (solid) and the algorithm from [10] (dashed) for system (49)-(50), feedback law $u = [-0.5 \ -0.3]x$ and prediction horizon $N = 4$ starting from different initial states close to the border of the feasible region of the latter algorithm. $[A_2 \ B_2]$ was used as plant model.

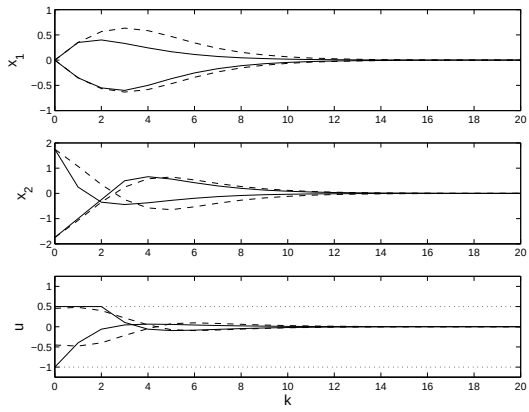

Figure 7: Input and state trajectories for for Algorithm 5 (solid) and the algorithm from [10] (dashed) for system (49)-(50), feedback law $u = [-0.5 \ -0.3]x$ and prediction horizon $N = 4$ starting from initial conditions $x(0) = [1.75 \ 0]^{\mathrm{T}}$ and $x(0) = [-1.75 \ 0]^{\mathrm{T}}$. $[A_2 \ B_2]$ was used as plant model. It is clear that Algorithm 5 can cope with non-symmetrical constraints more efficiently than the algorithm from [10].

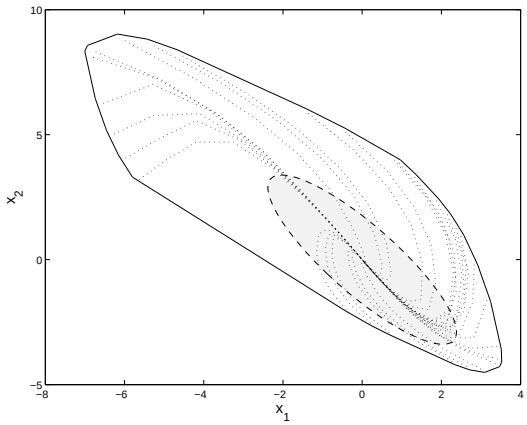

Figure 8: Trajectories for Algorithm 5 for system (49)-(50), feedback law $u = [-0.5 \ -0.3]x$ and prediction horizon $N = 4$ starting from different initial states close to the border of the feasible region. $[A_2 \ B_2]$ was used as plant model.

the ellipsoidal invariant sets. So, for completeness, Figure 8 shows that Algorithm 5 is robustly stabilizing within its entire feasible region.

Although theoretically Algorithm 5 has a lower computational complexity due to the fact that it is based on QP instead of SDP, it has the disadvantage that the number of constraints describing $\mathcal{S}_N$ can become impractically large for larger prediction horizons (in this example 13, 24, 46, 86 and 161 constraints for $N = 0, \ldots, 4$ respectively). This might seem to conflict with the fact that Algorithm 2 should be able to remove a large part of these constraints. However, as the dimensionality of the (augmented) system increases, the fraction of redundant constraints stays virtually the same due to the shift structure present in (42), leading to an exponential increase of the number of constraints as a function of N. Despite this disadvantage no noticeable computational performance degradations could be observed for the prediction horizons considered in this example.

7 Conclusion

In this paper an efficient algorithm for the construction of the maximal admissible set (MAS) for LPV systems subject to linear constraints is introduced. The new algorithm makes use of a new

invariance condition and avoids the combinatorial explosion of the number of constraints by removing redundant constraints in intermediate iterations. Possible extensions toward bounded additive disturbances and contraction constraints are also discussed. Secondly, the polyhedral invariant sets constructed using the new algorithm are applied to a robust MPC algorithm to increase feasibility, control performance and computational efficiency. The new algorithms introduced in this paper are illustrated using a numerical example.

Future research directions include the extension of the polyhedral invariant sets synthesis algorithm to continuous time systems. Also, new algorithms that allow the construction of invariant sets with a reduced number of constraints is a potentially useful research topic which is essential to enable the extension of polyhedral invariant sets towards large dimensional systems. Furthermore, modifications to the existing algorithm that allow the construction of minimal admissible sets or maximal control-admissible sets might have a large impact on robust model predictive control and robust constrained control in general.

Acknowledgements

Research supported by Research Council KULeuven: GOA-Mefisto 666, 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), research communities (ICCoS, ANMMM); AWI: Bil. Int. Collaboration Hungary/ Poland; IWT: PhD Grants, Soft4s (softsensors), Belgian Federal Government: DWTC (IUAP IV-02 (1996-2001) and IUAP V-22 (2002-2006)), PODO-II (CP/40: TMS and Sustainibility); EU: CAGE; ERNSI; Eureka 2063-IMPACT; Eureka 2419-FliTE; Contract Research/agreements: Data4s, Electrabel, Elia, LMS, IPCOS, VIB. Bert Pluymers is a research assistant with the I.W.T. (Flemish Institute for Scientific and Technological Research in Industry) at the Katholieke Universiteit Leuven. Dr. Johan Suykens is an associate professor at the Katholieke Universiteit Leuven, Belgium. Dr. Bart De Moor is a full professor at the Katholieke Universiteit Leuven, Belgium.

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] H. H. J. Bloemen, T. J. J. van den Boom, and H. B. Verbruggen. Optimizing the end-point state-weighting matrix in model-based predictive control. *Automatica*, 38:1061–1068, 2002.

[4] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM Publications, 1994.

[5] E.F. Camacho and C. Bordóns. *Model Predictive Control in the Process Industry*. Springer-Verlag, 1995.

[6] M. Cannon, V. Deshmukh, and B. Kouvaritakis. Nonlinear model predictive control with polytopic invariant sets. *Automatica*, 39:1487–1494, 2003.

[7] E. G. Gilbert and Tan. K. T. 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.

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

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

[10] B. Kouvaritakis, J.A. Rossiter, and J. Schuurmans. Efficient robust predictive control. *IEEE Transactions on Automatic Control*, 45(8):1545–1549, 2000.

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

[12] J.M. Maciejowski. *Predictive Control with Constraints*. Springer-Verlag, 2001.

[13] W.-J. Mao. Robust stabilization of uncertain time-varying discrete systems and comments on "an improved approach for constrained robust model predictive control". *Automatica*, 39:1109–1112, 2003.

[14] 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.

[15] 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.

[16] B. Pluymers, J.A. Rossiter, J. Suykens, and B. De Moor. A simple algorithm for robust MPC. In *Proceedings of the IFAC World Congress 2005, Prague, Czech Republic*, 2005.

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

[18] 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.

[19] P. O. M. Scokaert and D. Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998.

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

[21] Z. Wan and M. V. Kothare. Efficient robust constrained model predictive control with a time varying terminal constraint set. *Systems and Control Letters*, 48:375–383, 2003.