Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 05-134

Extending the Feasibility of Interpolation based MPC and Application to LPV Systems [1]

J.A. Rossiter[2] , B. Pluymers[3] , Y. Ding[2], X. Lei[2],
J.A.K. Suykens[3] and B. De Moor[3]

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/aut05_improved_gimpc.pdf*

[2]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*

[3]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.

Abstract

Interpolation techniques are known, in some cases, to reduce the computational complexity of MPC algorithms (Bacic *et al.*, IEEE TAC, 2003), (Rossiter *et al.*, IJC, 2004). However, this can be at a cost to the volume of the feasible region. This paper gives a new insight on general interpolation which allows a substantial increase in feasible regions, while retaining guaranteed convergence, and therefore the potential of the approach. Moroever, a further significant advance is an extension to the cases of linear parameter varying systems using polyhedral rather than elllipsoidal invariant sets.

Extending the feasibility of interpolation based MPC and application to LPV Systems

J.A. Rossiter‡, B. Pluymers†, Y. Ding‡, X. Lei‡, 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/`

May 31, 2005

Abstract

Interpolation techniques are known, in some cases, to reduce the computational complexity of MPC algorithms [1], [22]. However, this can be at a cost to the volume of the feasible region. This paper gives a new insight on general interpolation which allows a substantial increase in feasible regions, while retaining guaranteed convergence, and therefore the potential of the approach. Moroever, a further significant advance is an extension to the cases of linear parameter varying systems using polyhedral rather than elllipsoidal invariant sets.

1 Introduction

Predictive control (MPC) [5, 23] has been successfully applied to many real systems, including MIMO systems, because it builds constraint handling in from the outset. This is usually achieved by embedding an online optimization at each sampling instant, but with the consequent computational requirements. Herein lies a conflict: an MPC algorithm is expected to behave well within its feasible region (e.g. [6]), that is the operating region within which constraints are predicted to be satisfied, but the volume of this region is implicitly linked to the tuning and the number of degrees of freedom (d.o.f.) in the optimization. Hence, a necessary compromise is between achievable performance and the volume of the feasible region.

The compromise can be best elucidated by a simple example. For a stable system with only input constraints, one can always satisfy constraints by a reduction in gain [7]. This detuned control law consequently gives a large feasible region. However, better tuning can be achieved by a time varying control law which gradually retunes as the state moves closer to target [27, 17]. Predictive control can give both of these through the embedded d.o.f. [26] and a dual mode paradigm [12], but the size of the feasible region is linked to the number of d.o.f. and the terminal control. Notably, if the terminal control is well tuned, one may need prohibitive numbers of d.o.f. [20]. Hence there is a need for algorithms which can retain the properties of large feasible regions, optimal performance when near the target but also do not give rise to demanding optimizations; that is, they use only a few d.o.f. [21, 19].

This paper concentrates on one approach to this problem, that is interpolation. Initial work used polyhedral invariant sets [20] and showed how one could get substantial gains in feasibility while deploying just one d.o.f., however, stability and convergence results [22] are weak. Moreover, these results apply only to the nominal case. Parallel work [1, 17] explored the use of ellipsoidal invariant sets; this had the advantage of being applicable to uncertain systems, but at the price of relatively poor feasibility and suboptimal performance.

This paper makes two significant contributions. First it shows new insights by combining ideas in [20] and [1] and hence proposes a new interpolation with a significantly increased feasible region. Secondly, it makes use of new results on robust polyhedral invariant sets [14] and shows how these can be modified and then employed in the new interpolation which, unlike earlier results, can therefore be applied to the uncertain case, hence improving on the restricted feasibility of earlier works.

Section 2 will give necessary background to MPC and invariant sets. Section 3 will first give a critique and hence new insights into interpolation before then proposing a new interpolation algorithm for the nominal case. Section 4 works through the technical details to apply this to the uncertain case. Section 5 gives some numerical examples.

2 Background

2.1 Model and objective

In this paper we will consider both linear time invariant (LTI, $m = 1$) and linear parameter-varying (LPV, $m > 1$) systems of the form

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

with

$$[A(k)\ B(k)] \in \Omega \triangleq \mathrm{Co}\{[A_1\ B_1], \ldots, [A_m\ B_m]\}, \tag{1b}$$

subject to constraints

$$u(k) \in \mathcal{U} \equiv \{u : \underline{u} \leq u \leq \overline{u}\}, \qquad k = 0, \ldots, \infty, \tag{2a}$$
$$x(k) \in \mathcal{X} \equiv \{x : \underline{x} \leq x \leq \overline{x}\}, \qquad k = 0, \ldots, \infty. \tag{2b}$$

$x(k) \in \mathbb{R}^{n_x}$ and $u(k) \in \mathbb{R}^{n_u}$ denote state and input vectors at discrete time k. For notational simplicity, only component-wise peak bound are considered, but more general linear (mixed) state and input constraints can also be included. In what follows, when dealing with LTI models, we will talk about the *nominal case*, while when dealing with LPV models, we will talk about the *uncertain* or *robust case*.

Problem 1 (Cost Objective) *The aim of this paper is to formulate an interpolation-based control algorithm that robustly stabilizes system* (1) *and guarantees robust satisfaction of constraints* (2) *based on given asymptotically stabilizing feedback controllers* $K_1, \ldots, K_n$ *. The algorithm should aim to minimize an upper bound (over all possible system dynamics satisfying the uncertainty description) on*

$$J = \sum_{k=0}^{\infty} (x(k)^{\mathrm{T}} Q x(k) + u(k)^{\mathrm{T}} R u(k)), \tag{3}$$

as a cost objective with $Q = Q^{\mathrm{T}} > 0 \in \mathbb{R}^{n_x \times n_x}$ *and* $R = R^{\mathrm{T}} > 0 \in \mathbb{R}^{n_u \times n_u}$ *state and input cost weighting matrices.*

Remark 1 *In this and future sections, when talking about the feedback law* K *the following feedback law is implicitly assumed :*

$$u(k) = -Kx(k); \quad \forall k. \tag{4}$$

2.2 Invariant Sets

Invariant sets [3] are key to the developments discussed in this paper and hence are overviewed next.

Definition 1 (Feasibility and robust positive invariance) *Given a system, stabilizing feedback and constraints (1,4,2), then a set* $\mathcal{S} \subset \mathbb{R}^{n_x}$ *is feasible iff* $\mathcal{S} \subseteq \{x | x \in \mathcal{X}, -Kx \in \mathcal{U}\}$. *Moreover, the set is robust positive invariant iff*

$$x \in \mathcal{S} \quad \Rightarrow \quad (A - BK)x \in \mathcal{S}, \quad \forall [A \ B] \in \Omega. \tag{5}$$

Remark 2 Positive *invariance signifies that only future state predictions should lie within the invariant set (and not past states). However, for simplicity of notation in further sections positive invariance will be referred to as invariance.*

Remark 3 *The largest feasible invariant set (in the sense that no other feasible invariant set can contain states outside this set) is uniquely defined and is called the Maximal Admissable Set (MAS, [6]).*

2.2.1 Polyhedral invariant sets for the nominal case

Define the closed-loop predictions for a given feedback K as:

$$x(k) = \Phi^k x(0); \quad u(k) = -K\Phi^{k-1}x(0); \quad \Phi = A - BK. \tag{6}$$

Then, under mild conditions [6] the MAS for a controlled LTI system is given by

$$\mathcal{S} = \bigcap_{k=0}^{n} \{x | \Phi^k x \in \mathcal{X}, -K\Phi^k x \in \mathcal{U}\} = \{x : Mx \leq d\}, \tag{7}$$

with n a finite number; n being finite indicates that $\mathcal{S}$ is polyhedral. Hereafter, assume that $x = 0$ is in the interior and hence normalise inequalities (7) to take the form $d = [1, 1, ..., 1]^{\mathrm{T}}$ and introduce the shorthand notation

$$\lambda S \equiv \{x : Mx \leq \lambda d\}. \tag{8}$$

2.2.2 Ellipsoidal and polyhedral invariant sets for the LPV case

In the LPV case, an invariant set must exist if a quadratic stabilizability condition is satisfied. That is, there exists a solution $P_i = P_i^{\mathrm{T}} > 0 \in \mathbb{R}^{n_x \times n_x}$ for a feedback K_i to:

$$\Phi_{ij}^{\mathrm{T}} P_i \Phi_{ij} \leq P_i, \quad \forall j, \quad \Phi_{ij} = A_j - B_j K_i. \tag{9}$$

A corresponding invariant set of ellipsoidal form is $\mathcal{S} = \{x | x^{\mathrm{T}} P_i x \leq 1\}$. Suitable P_i can be calculated using results such as those in [9], by solving an SDP. The relative ease of computing suitable P_i has facilitated widespread use of ellipsoidal invariant sets, however, their restrictive shape and complexity can imply significant conservatism, notably with non-symmetrical constraints [15, 16].

Recent developments [14] have demonstrated the tractability of algorithms to compute the MAS (polyhedral invariant sets) for an LPV system. The key idea [8] avoids the explicit

predictions of (7) and hence the possible combinatorial explosion in the number of prediction terms [25]. The reader is referred to [14] for details and efficient implementations; here a brief summary followed by outline algorithm of the conceptual idea is given. The algorithm searches iteratively, beginning with an outer estimate, for an $\mathcal{S} = \{x : Mx \leq d\}$ such that $x \in \mathcal{S} \;\; \Rightarrow \Phi x \in \mathcal{S}, \; \forall \Phi$. If the current outer estimate for $\mathcal{S}$ does not satisfy this test, then it is augmented with additional rows $M\Phi, \; \forall \Phi$ and the test is repeated. Condition (9) ensures convergence in finite time.

Algorithm 1 *Perform the following steps :*

1. *For a given feedback K, assume that an outer approximation to the MAS is given by $\mathcal{S}_0 = \{x : M_0 x \leq d_o\} \equiv \{x : x \in \mathcal{X}, \; u = -Kx \in \mathcal{U}\}$.*
2. *Set up an iteration on sets $\mathcal{S}_k$, initialized with $\mathcal{S}_0$, such that we find the sets $\mathcal{S}_{k-1}$ of previous states $x(-1)$ such that $x(0) \in \mathcal{S}_k$ to give*
$$\mathcal{S}_{k-1} = \{x : \; x(-1) \in \mathcal{S}_k, \; [A - BK]x(-1) \in \mathcal{S}_k, \forall (A, B) \in \Omega\}. \tag{10}$$
3. *Write the current sets in the following form*
$$\mathcal{S}_k = \{x : M_k x \leq d_k\}. \tag{11}$$
4. *Remove from matrices $M_k, \; d_k$ rows corresponding to constraints that are redundant.*
5. *Return to step 2 and iterate until $\mathcal{S}_{k-1} \equiv \mathcal{S}_k$. Remove subscripts 'k' to give the MAS as $\mathcal{S} = \{x : Mx \leq d\}$.*

2.3 General Interpolation [1]

Assume there are a number of alternative stabilizing feedbacks $K_i, i = 1, \ldots, n$ of which K_1 is the preferred choice and that $\Phi_{ij} = A_j - B_j K_i$. Then, given these, a system (1), constraints (2), corresponding invariant sets $\mathcal{S}_i, i = 1, \ldots, n$ and a current state $x(0)$, define the following decomposition

$$x(0) = \sum_{i=1}^{n} x_i, \quad \text{with} \quad \left\{ \begin{array}{l} \sum_{i=1}^{n} \lambda_i = 1, \; \lambda_i \geq 0, \\ x_i \in \lambda_i \mathcal{S}_i, \end{array} \right. \tag{12}$$

with associated input/state predictions

$$\begin{array}{l} u(k) = -\sum_{i=1}^{n} K_i \prod_{j=0}^{k-1} \Phi_i(k-1-j) x_i, \\ x(k) = \sum_{i=1}^{n} \prod_{j=0}^{k-1} \Phi_i(k-1-j) x_i, \end{array} \tag{13}$$

and feasible region $\overline{\mathcal{S}}$

$$\overline{\mathcal{S}} \triangleq \mathrm{Co}\{\mathcal{S}_1, \ldots, \mathcal{S}_n\}. \tag{14}$$

These decomposition/predictions are recursively feasible, that is $x(k) \in \overline{\mathcal{S}} \;\Rightarrow\; x(k+1) \in \overline{\mathcal{S}}$. Lyapunov theory is used to compute an upper bound $\tilde{x}^{\mathrm{T}} P \tilde{x}$ on the infinite-horizon cost J of (3) where $\tilde{x} = [\hat{x}_1^{\mathrm{T}} \;\; \ldots \;\; \hat{x}_n^{\mathrm{T}}]^{\mathrm{T}}$ and $P = P^{\mathrm{T}} > 0 \in \mathbb{R}^{(n \cdot n_x) \times (n \cdot n_x)}$ satisfies

$$P \geq \Gamma_u^{\mathrm{T}} R \Gamma_u + \Psi_i^{\mathrm{T}} \Gamma_x^{\mathrm{T}} Q \Gamma_x \Psi_i + \Psi_i^{\mathrm{T}} P \Psi_i, \quad i = 1, \ldots, m, \tag{15}$$

with $\Psi_i = [\Phi_{1i}^{\mathrm{T}} \;\; \ldots \;\; \Phi_{ni}^{\mathrm{T}}]^{\mathrm{T}}$, $\Gamma_x = [I, \ldots, I]$, $\Gamma_u = [K_1, \ldots, K_n]$.

Algorithm 2 (GIMPC: MPC using General Interpolation) *Take a system* (1)*, constraints* (2)*, cost weighting matrices* Q, R*, controllers* K_i *and ellipsoidal invariant sets* $\mathcal{S}_i$ *and compute a suitable* P *from (15). Then, at each time instant, given the current state (for simplicity of notation denoted as* $x(0)$*), solve the following optimization:*

$$\min_{\hat{x}_i, \lambda_i} \ \tilde{x}^{\mathrm{T}} P \tilde{x}, \quad \textit{subject to } (12), \tag{16}$$

and implement the input $u = -\sum_{i=1}^{n} K_i x_i$*.*

Algorithm 2 has guarantees of recursive feasibility and asymptotic stability and comprises the mode 2 only algorithm from [1]. However, the existing literature uses only ellipsoidal invariant sets and extensions to allow the use of polyhedral robust invariant sets remain to be developed.

2.4 One degree of freedom interpolations [20, 22]

Next, a brief overview of the simplest interpolation algorithms [22] is required to demonstrate new insights and as a building block for the proposed developments. ONEDOF algorithms deploy just one d.o.f. to interpolate between two possible feedbacks K_1, K_2 (extensions to more K_i are possible). This is enabled by a restriction to co-linear interpolation which means that unlike (12), the current state is decomposed as follows:

$$x = x_1 + x_2; \quad x_1 = (1-\alpha)x; \quad x_2 = \alpha x; \quad 0 \le \alpha \le 1. \tag{17}$$

If $x \in \mathcal{S}_1$, then one would always choose $\alpha = 0$, but otherwise two different versions exist:

Algorithm 3 *[ONEDOFa] The first control move is* $u = -[(1-\alpha)K_1 + \alpha K_2]x$ *where* α *is determined from:*

$$\min_{\alpha} \quad \alpha \ \text{ s.t. } \ [M_1(1-\alpha) + M_2\alpha]x \le d; \quad 0 \le \alpha \le 1. \tag{18}$$

Algorithm 4 *[ONEDOFb] The first control move is* $u = -[(1-\alpha)K_1 + \alpha K_2]x$ *where* α *is determined from:*

$$\min_{\alpha,\beta} \quad \alpha \ \text{ s.t. } \ \begin{cases} M_1(1-\alpha)x \le (1-\beta)d, \\ M_2\alpha x \le \beta d, \\ 0 \le \beta \le 1; \quad 0 \le \alpha \le 1. \end{cases} \tag{19}$$

This is known to be solved by $\alpha = (\mu - 1)/(\mu - \lambda)$ *where* $\mu = \max(M_1 x), \quad \lambda = \max(M_2 x)$*.*

The feasible regions $\mathcal{S}_{O1}$ and $\mathcal{S}_{O2}$ for (18,19) respectively are:

$$\begin{array}{rcl} \mathcal{S}_{O1} & = & \{\mathbf{x} : \exists \alpha \text{ s.t. } [M_1(1-\alpha) + M_2\alpha]\mathbf{x} \le d\}, \\ \mathcal{S}_{O2} & = & \{\mathbf{x} : \exists \alpha, \ \beta \ \text{ s.t. } \ M_1(1-\alpha)\mathbf{x} \le (1-\beta)d, \ M_2\alpha\mathbf{x} \le \beta d\}. \end{array} \tag{20}$$

Remark 4 *The weakness of ONEDOF is the difficulty of establishing recursive feasibility and convergence. The latter requires a condition similar to (9) and the former has only been proven for ONEDOFb because condition (19) is analogous to (12).*

3 New insights to interpolation

It is originally thought [1] that when combining two control laws, it was not unreasonable to expect the resulting feasible region to be the convex hull $\overline{\mathcal{S}}$ of the underlying MAS. However, a case study [22] contradicted this as the feasible region for ONEDOFa included points outside of $\overline{\mathcal{S}}$; something general interpolation cannot because $x \in \overline{\mathcal{S}}$ is enforced through (12). However, [22] did not pursue this observation and moreover $\mathcal{S}_{O1}$ was non-convex.

The observation that the feasible region for a control law constructed as a simple mix of underlying laws may be far larger than the convex union of the underlying MAS for these laws is both novel and potentially very useful. It implies one may be able to achieve larger feasible regions than originally anticipated from interpolation techniques, hence this paper seeks to understand and build on it.

The following subsections discuss these new insights for the LTI case, whereas the next section discusses the LPV case.

3.1 Understanding the conservatism in ONEDOF and GIMPC

ONEDOFb is analogous to GIMPC, (apart from the additional co-linearity condition) in that constraints (12,19) both imply

$$x_1 \in (1-\beta)\mathcal{S}_1;\ x_2 \in \beta\mathcal{S}_2;\ 0 \leq \beta \leq 1. \tag{21}$$

Lemma 1 *ONEDOFb has conservative feasibility, that is $x \in \overline{\mathcal{S}}$ implies the predictions are feasible but $x \notin \overline{\mathcal{S}}$ need not imply constraint violations.*

Proof: Conditions (21) compute the worst case for each $x_i(k), u_i(k)$ independently and ensure the sum is less than 100%. As the respective worst cases for $x_1,\ x_2$ may occur at different k, the actual trajectories x may not go close to 100%. So for instance, using a single state example with just an upper constraint on x

$$\min_{\alpha} \alpha \ \text{ s.t. } \ \max_k[\Phi_1^k x](1-\alpha) + \max_j[\Phi_2^j x]\alpha \leq \overline{x};\ \ 0 \leq \alpha \leq 1, \tag{22}$$

then, if $k \neq j$, necessarily $\max_k[\Phi_1^k x(1-\alpha) + \Phi_2^k x\alpha] < \overline{x}$. □

Lemma 2 *ONEDOFa does explicit testing of predictions against constraints.*

Proof: Using the same example as above and considering then the k-th row

$$\min_{\alpha} \alpha \ \text{ s.t. } \ \max_k e_k^{\mathrm{T}}[M_1(1-\alpha) + M_2\alpha]x \leq \overline{x}. \tag{23}$$

This is equivalent to $x(k) = x_1(k) + x_2(k) = [\Phi_1^k(1-\alpha) + \Phi_2^k\alpha]x \leq \overline{x}$. The definitions of M_i (7) and the formulations of constraints in (18,19), show that the statement applies more generally. □

In summary, ONEDOFa does explicit constraint checking and hence gives the largest possible feasible region for the given class of predictions. However, there is no recursive feasibility proof. ONEDOFb on the other hand adds the maximum for each individual component along the whole trajectory giving restricted but recursive, feasibility ($\mathcal{S}_{O2} \subseteq \mathcal{S}_{O1}$, [22]).

Lemma 3 *Matrices $M_1,\ M_2$ within ONEDOFa must have identical structures, that is the k-th row of each must be associated to an identical prediction/constraint. As a consequence, the matrices $M_1,\ M_2$ may be non-minimal (i.e., contain redundant constraints).*

Proof: This first statement is an obvious consequence of the previous 2 lemmas. The second statement arises because usual algorithms to form MAS remove redundant constraints. However, although the k-th row maybe redundant in $\mathcal{S}_1$, it may not be in $\mathcal{S}_2$, and hence the k-th row must be retained in both M_1 and M_2. In fact, the observation goes further still. Both the k-th rows of $\mathcal{S}_1,\ \mathcal{S}_2$ may be redundant when taken individually, but are not so when taken together as in (23). □

The following Theorem is given without proof, which is obvious from the preceding lemmata.

Theorem 1 *ONEDOFa has larger feasible regions than ONEDOFb, in general, but ONEDOFb requires less inequalities.*

Corollary 1 *In order to define $\mathcal{S}_{O1}$ it is necessary to compute matrices $M_1,\ M_2$ in parallel to ensure constraints are removed as redundant only if they are redundant as measured by the inequalities in (23).*

3.2 Interpretations of general interpolation

ONEDOF algorithms suffer from weak convergence proofs, due to the collinearity condition, a fact GIMPC overcomes by allowing that $x_1(k+1) = \Phi_1 x_1(k)$, $x_2(k+1) = \Phi_2 x_2(k)$ hence enabling a simple proof of convergence and recursive feasibility. Furthermore GIMPC extended the guaranteed feasible region from $\bigcup_i \mathcal{S}_i$ to $\overline{\mathcal{S}}$. However, GIMPC[1] still requires condition (21) to ensure recursive feasibility and hence includes the conservatism discussed in lemma 1. In fact, so much so that at times ONEDOFa may even have better feasibility than GIMPC [22].

The proposal here is to use the d.o.f. of general interpolation which allow convergence and recursive feasibilty results, but combine this with constraints of the form of $\mathcal{S}_{O1}$ and hence improve feasible regions by removing conservatism from the constraint handling. This is straightforward for the LTI case and is presented next. The LPV case requires some technical developments given in section 4.

[1]The use of invariant ellipsoids (especially for non-symmetric constraints) instead of the MAS is also conservative but this issue was tackled in [22], albeit only for the LTI case.

3.3 Improved general interpolation: GIMPC2

The proposed algorithm GIMPC2 relates to GIMPC analogously to ONEDOFa relating to ONEDOFb, that is we remove the condition (21) or more generally condition

$$\begin{cases} x_i \in \lambda_i \mathcal{S}_i, \quad 0 \leq \lambda_i \leq 1, \quad i = 1, \ldots, n, \\ \sum_{i=1}^{n} \lambda_i = 1, \end{cases} \tag{24}$$

introduced in [1] to enforce recursive feasibility. These are replaced solely by (2) on the actual predictions but, because of the use of MAS, this change does not cause a loss of the important recursive feasibility property. Hence, define the feasible regions for GIMPC2 and GIMPC respectively as:

$$\begin{array}{rcl} \mathcal{S}_{G2} & = & \{x : \exists x_1, x_2, \ldots, x_n, \text{ s.t. } M_1 x_1 + M_2 x_2 + \ldots M_n x_n \leq d, x = \sum_{i=1}^{n} x_i\}, \\ \mathcal{S}_{G} & = & \{x : \exists x_1, x_2, \ldots, x_n, \text{ s.t. } M_i x_i \leq \lambda_i d, \sum_{i=1}^{n} \lambda_i = 1, \sum_{i=1}^{n} x_i = x, \ \lambda_i \geq 0, \ i = 1, \ldots, n\}. \end{array} \tag{25}$$

The following two lemmata are obvious (following on from the previous section), and are therefore stated without proof.

Lemma 4 *The feasible region $\mathcal{S}_G$ for GIMPC is a subset of $\mathcal{S}_{G2}$, the region for GIMPC2.*

Lemma 5 *When formulating the inequalities $M_i x_i \leq d_i$, $\forall i$ for GIMPC2, there is a requirement to check redundancy against the combined inequalities of (25a). Specifically, the combined constraints take the form:*

$$[M_1, M_2, \ldots, M_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \leq d, \quad x = \sum_{i=1}^{n} x_i, \tag{26}$$

with M_i defined as in (7) and normalised so that the corresponding $d_i = [1, 1, \ldots, 1]^{\mathrm{T}}$.

Algorithm 5 (GIMPC2) *Using the same notation as algorithm 2, at each time instant, given the current state x, solve the following optimization problem on-line*

$$\min_{x_i} \ \tilde{x}^{\mathrm{T}} P \tilde{x}, \quad \textit{subject to} \quad \begin{cases} \sum_{i=1}^{n} M_i x_i \leq d, \\ x = \sum_{i=1}^{n} x_i, \end{cases} \tag{27}$$

and implement the input $u = -\sum_{i=1}^{n} K_i x_i$.

Theorem 2 *Algorithm 5 has guaranteed convergence and recursive feasibility.*

Proof: These follow well known arguments. Because the constraint equation is based on trajectories (see (7)) and from Lemma 5 all possibly active predictions are included, hence

$$\sum_{i=1}^{n} M_i x_i \leq d \quad \Rightarrow \quad \sum_{i=1}^{n} M_i \Phi_i x_i \leq d. \tag{28}$$

Recursive feasibility is then immediate if one allows the choices $x_i(k+1) = \Phi_i x_1(k)$, $\forall i$. Secondly, because of the use of infinite horizons in J, these choices of x_i [23] are also sufficient to establish that J is Lyapunov and hence convergence is proven. □

Remark 5 *The following remarks can be made about the new algorithm :*

- *The reason why a simple mix of control laws can give such benefits is because the class of controllers is in fact far larger than simply $K = (1-\alpha)K_1 + \alpha K_2$, because of the flexiblity in the state decomposition. Moreover, the implied strategy within the predictions is time-varying (e.g. (13)).*
- *GIMPC2 uses fewer variables than GIMPC as the λ_i variables are not needed.*
- *GIMPC uses fewer inequalities than GIMPC2 because each respective set $\mathcal{S}_i$ can be reduced to minimal form.*

4 The robust case

The previous section dealt solely with the nominal case. However, one main focus of general interpolation [1] and current research interest is the robust or LPV case. Earlier works have relied on ellipsoidal invariant sets so this section shows how the GIMPC2 algorithm, which is only defined for polyhedral sets, can be extended to the LPV case. In fact the main requirement is very simple. Algorithm 5 is applicable to the LPV case in identical format, as long as matrices, P, M_i, d_i are defined appropriately. The next section presents algorithms for doing this.

4.1 Defining mutually consistent polyhedral invariant sets

The algorithm of [14] was defined to find the MAS of an LPV system for a single control law. Due to issues of possible combinatorial explosion in the number of inequalities, it was necessary to include within the iteration frequent checks for and removal of redundant constraints. However, as noted in lemmas 3, 5, for the GIMPC2 algorithm, constraints may need to be retained even where they are redundant in all the individual MAS. For the LPV case, due to combinatorial explosions, one can not simply define the MAS as in (7) and then remove redundant constraints afterwards. Hence an efficient algorithm to deal with this complexity is required. Here one possibility, forming an augmented system, is discussed.

Algorithm 6 (Method to find mutually consistent MAS for the LPV case)

1. *Define an augmented system*

$$X(k+1) = \Psi(k)X(k); \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \tag{29}$$

$$\Psi(k) = \begin{bmatrix} A(k) - B(k)K_1 & 0 & \dots & 0 \\ 0 & A(k) - B(k)K_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A(k) - B(k)K_n \end{bmatrix}.$$

Define a set $\Psi \in \hat{\Omega}$, describing the allowable variation in Ψ due to the variations implied by $[A(k)\ B(k)] \in \Omega$.

2. *Constraints (2) need to be written in terms of augmented state X as follows:*

$$\{\underline{u} \leq \underbrace{[-K_1, -K_2, \cdots]}_{\hat{K}} X(k) \leq \overline{u}\}, \qquad k = 0, \dots, \infty, \tag{30a}$$

$$\{\underline{x} \leq [I, I, \cdots] X(k) \leq \overline{x}\}, \qquad k = 0, \dots, \infty. \tag{30b}$$

3. *Assume that an outer approximation to the MAS is given by (30). Then letting $u = -\hat{K}X$, this reduces to $\mathcal{S}_o = \{X : M_o X \leq d_o\}$ where definitions of M_o, d_o are obvious.*

4. *Follow the steps 2-5 of Algorithm 1 to find the robust MAS as $\mathcal{S}_a = \{X : M_a X \leq d_a\}$ (subscript 'a' for augmented).*

Remark 6 (Feasible region for robust GIMPC2) *Given the constraint $x = \sum_{i=1}^{n} x_i$, then one can find a projection of $\mathcal{S}_a$ to x-space from X-space as follows:*

$$\mathcal{S}_{G2} = \{x : x = [I, I, \dots, I]X,\ \exists X \text{ s.t. } M_a X \leq d_a\}. \tag{31}$$

Remark 7 *One can identify the underlying MAS, $\mathcal{S}_i$ from $\mathcal{S}_a$ by setting $x_j = 0,\ j \neq i$ in turn, that is $M_1 = M_a[I, 0, 0, \dots, 0]^{\mathrm{T}},\ M_2 = M_a[0, I, 0, \dots, 0]^{\mathrm{T}},\ d_1 = d_2 = d_a$. However, these M_i will contain many redundant rows.*

Remark 8 *There are alternative ways of forming an augmented system/states using the observation that $x = \sum x_i \quad \Rightarrow \quad x_r = x - x_1 - x_2 - ... - x_{r-1}$.*

4.2 GIMPC2 for the LPV case

The basic suggestion hereafter is unsurprisingly quite simple. Take the polyhedral invariant sets discussed in algorithm 6 and use these in a GIMPC2 algorithm 5.

Algorithm 7 (GIMPC2 for the LPV case) *Given a system (1), constraints (2), cost weighting matrices $Q = Q^{\mathrm{T}} > 0, R = R^{\mathrm{T}} > 0$, asymptotically stabilizing controllers K_i, corresponding polyhedral robust invariant sets $\mathcal{S}_a = \{X : M_a X \leq d_a\}$ and P satisfying (15), solve on-line at each time instant, given the current state x, the following problem :*

$$\min_{\hat{x}_i} \tilde{x}^{\mathrm{T}} P \tilde{x}, \textit{ subject to } \quad \begin{cases} x = [I, I, \dots, I]X, \\ M_a X \leq d_a, \end{cases} \tag{32}$$

and implement input $u = -[K_1,\ K_2,\ \dots,\ K_n]X$.

Theorem 3 *Algorithm 7 guarantees robust satisfaction of* (2) *and is recursively feasible and asymptotically stable for all initial states* $x(0) \in \mathcal{S}_{G2}$.

Proof: This follows similar lines to Theorem 2. The only change is the requirement to show that, irrespective of the process values $A(k), B(k)$ (or $\Psi(k)$), that nevertheless

$$x(k) \in \mathcal{S}_{G2} \quad \Rightarrow \quad x(k+1) \in \mathcal{S}_{G2}. \tag{33}$$

This follows immediately from the invariance and feasibility of $\mathcal{S}_a$. Asymptotic stability also follows similar lines: it is known that one can always choose new state components at the next time step as those as suggested in (12). However, repeated choice of the same decomposition must give convergence due to the quadratic stability (9) associated to each individual control law, and hence also to augmented system Ψ. Deviation away from this scheme will only occur where the cost $J = \tilde{x}^{\mathrm{T}} P \tilde{x}$ can be made smaller still. This consequently proves that the optimal value of the cost function of (32) acts as a Lyapunov function of the closed-loop system, which proves asymptotic stability. □

4.3 Summary

For convenience we summarize the contributions of this paper with the associated properties. A key point is the change from implicit or worst case constraint handling in GIMPC to explicit constraint handling in GIMPC2.

Algorithm	Contribution
GIMPC	Extended application based on polyhedrals to the LPV case. Hence increased feasible regions well beyond the ellipsoidal approach of [1]. Reduced computation as removed need for an SDP solver.
GIMPC2	New general interpolation to increase feasibility beyond the convex hull. Showed how to use robust polyhedral sets of [15, 16] to extend to LPV case. Reduced online computation by removing constraint/variables: $\sum_i \lambda_i = 1,\ \lambda_i \geq 0$.

5 Numerical Example

In this section we use a double integrator example, extended with some uncertainty, to demonstrate advantages of the new algorithm. As the developments of GIMPC2 for the nominal case are subsumed by those for the LPV case, to save space, these will not be illustrated in detail.

5.1 Model and constraints

We consider the LPV model given by

$$A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{34a}$$

$$A_2 = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \tag{34b}$$

subject to constraints

$$\overline{u} = 1, \qquad \underline{u} = -0.5, \tag{35a}$$

$$\overline{x} = [8,\ 8]^{\mathrm{T}}, \qquad \underline{x} = [-10,\ -10]^{\mathrm{T}}. \tag{35b}$$

A nominal model is chosen as $A = 0.5(A_1 + A_2), B = 0.5(B_1 + B_2)$ as the design basis for the two feedback controllers; one the LQR-optimal controller $K = [0.4558\ 0.3698]^{\mathrm{T}}$ with $Q = \mathrm{diag}(1, 0.01), R = 3$; the second a sub-optimal, detuned controller $K = [0.5\ 0.3]^{\mathrm{T}}$. Both controllers are robustly asymptotically stabilizing for system (34).

For comparison purposes we will adopt as a benchmark the algorithm of [26] (denoted OMPC) but modified to make use of robust MAS [18]. This algorithm deploys n_c d.o.f. where the robust n_c-set can be interpreted as the points from which one can guarantee entry into the terminal region, that is $\mathcal{S}_1$, in at most n_c steps (for all models in (1)).

5.2 Feasible Regions

There are several aspects which need to be illustrated.

1. Figure 1 shows the reduction in the polyhedral feasible region brought about by taking the model uncertainty into account. This is discussed in more detail in [14, 18] as is the necessity of using robust sets, so not repeated here.

2. Figure 2 shows the increase in feasibility by moving from ellipsoidal invariant sets to polyhedral invariant sets. It is clear that the use of polyhedral sets is a considerable advantage, with both feedbacks, especially when handling asymmetry.

3. Figure 3 shows the advantage in moving to GIMPC2. The feasible region for GIMPC2 is far larger than that for GIMPC, even though the underlying sets $\mathcal{S}_1,\ \mathcal{S}_2$ are fairly similar.

4. Figure 4 shows the comparable feasible regions for a robust optimal control law [18] with n_c d.o.f. Once again it is clear that GIMPC2 has far larger feasibility, even compared to $n_c = 5$.

5.3 Computational Load

One might expect the computational load to increase significantly with the move from the nominal to the uncertain case. Moreover, given the marked improvements in feasibility, one might expect GIMPC2 to be computationally more demanding. Neither of these are true and in fact GIMPC2 has the fewest degrees of freedom, but the move from the nominal case to the LPV case does give far larger numbers of inequalities, although of course with great benefits to feasibility. A comparison of d.o.f. and numbers of inequalities, summarized in table 1, confirms these expectations.

We do not consider GIMPC based on ellipsoids as the optimisation is an SDP and therefore more demanding than a QP in general.

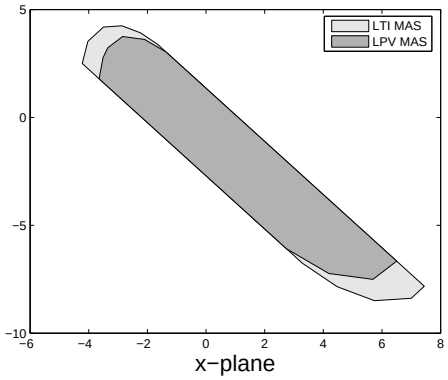


Figure 1: MAS for nominal (LTI) and LPV models (34)-(35) with $K = [0.4558\ 0.3698]^{\mathrm{T}}$.

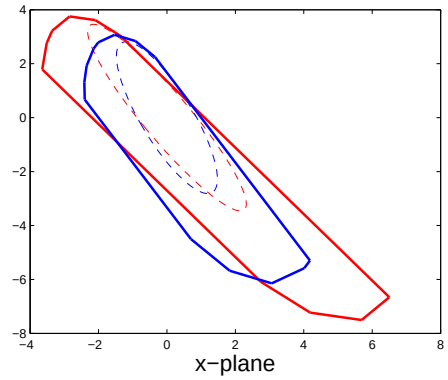


Figure 2: Feasible region comparison of polyhedral (solid line) and ellipsoidal invariant sets (dashed line) for feedback laws $K_1,\ K_2$ applied to model (34)-(35).

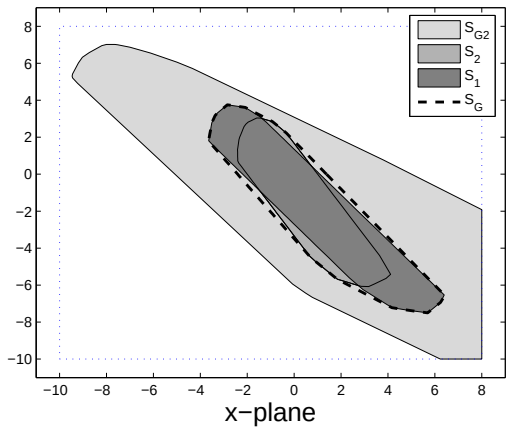


Figure 3: Feasible regions for algorithms GIMPC and GIMPC2 and the underlying MAS for model (34)-(35).

5.4 Control Performance

Illustrations are meaningful only if the initial condition is within the feasible region of all algorithms to be compared. So we compare GIMPC2 with GIMPC using initial conditions

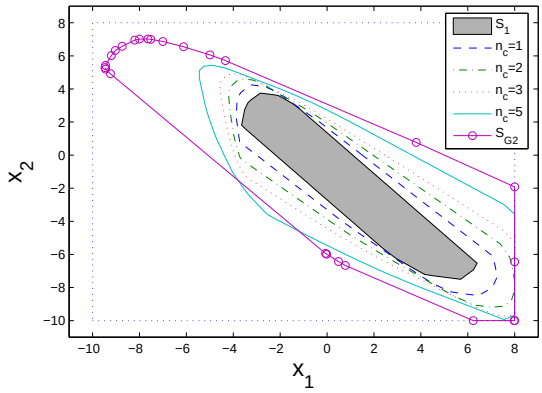


Figure 4: Feasible regions for algorithms GIMPC2 and OMPC (with various n_c) for model (34)-(35).

	Algorithm		
	GIMPC	GIMPC2	OMPC
No. inequalities	25	140	194
No. d.o.f.	$n_x + 1 = 3$	$n_x = 2$	$n_c = 5$

Table 1: Numbers of inequalities and degrees of freedom required by OMPC, GIMPC and GIMPC2 for model (34)-(35).

spread evenly around the feasibility boundary of GIMPC ($\mathcal{S}_G$) and we compare GIMPC2 with OMPC ($n_c = 5$) using points spread evenly around the feasibility boundary of OMPC (but also within $\mathcal{S}_{G2}$). The plant is varied randomly within the limits of (1) and we assume that by using a large number of simulations these effects will average out.

Simulation costs corresponding to cost J are computed, normalised and then averaged (table 2). Despite using fewer d.o.f. and having better feasibility, GIMPC2 has still performed as well as OMPC and better than GIMPC.

Algorithm		
GIMPC	GIMPC2	OMPC ($n_c = 5$)
1.02	1.00	1.00

Table 2: Normalised average runtime costs (3 s.f.) for GIMPC, GIMPC2 and OMPC for model (34)-(35).

5.5 Constraint Handling

Here we give evidence of the effectiveness of GIMPC2 in using all the control freedom available despite the asymmetry in the limits. Figure 5 presents input trajectories beginnning from boundary boints of $\mathcal{S}_{G2}$ (in fig (a)) and boundary points of $\mathcal{S}_G$ (in fig (b)). It is clear that GIMPC2 takes the inputs close to constraints for all points outside $\mathcal{S}_1$. This is significant because it is well known that (e.g. see [15, 16]) ellipsoidal GIMPC is very conservative, especially with non-symmetric constraints.

Polyhedral GIMPC may also be conservative for some examples, but the effect is expected to be relatively small in general, so not illustrated here.

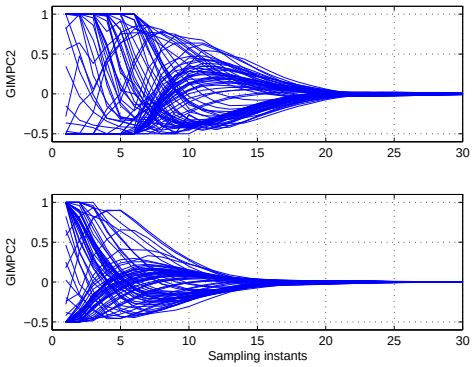


Figure 5: Input trajectories for GIMPC2 from initial conditions spread evenly around the boundaries of $\mathcal{S}_{G2}$ (upper figure) and $\mathcal{S}_G$ (lower figure) as depicted in Fig. 3

5.6 Robust closed-loop behaviour

Some evidence is also provided that GIMPC2 is indeed robustly feasible and convergent for the entire region displayed in figure 3. Closed-loop trajectories are computed for several points on the boundary and these are displayed in figure 6. Clearly they all both remain in $\mathcal{S}_{G2}$ and converge to the origin.

It is also interesting to note that the trajectories for GIMPC may be quite different to those from GIMPC2. In fact differences are small in this example and hence not plotted, but other examples (e.g. see [24]) show similar cases where the differences are large.

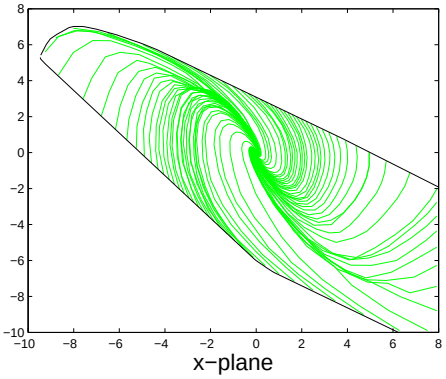


Figure 6: Closed-loop state trajectories for GIMPC2 applied to model (34)-(35).

6 Conclusions

A new interpolation based MPC algorithm using polyhedral robust invariant sets is introduced for LPV systems with polytopic model uncertainty. The new method makes use

of a new algorithm proposed in [14] that allows the efficient construction of polyhedral robustly invariant sets for LPV systems.

The two main contributions are :

- the extension of the general interpolation or GIMPC algorithm of [1] to make use of robust polyhedral invariant sets. This paves the way for more efficient handling of non-symmetric constraints as well as improving performance and feasibility even for the symmetric case.
- the development of a new interpolation algorithm (GIMPC2), applicable to nominal and robust cases, which extends feasibility far more widely than existing methods envisaged; in fact well beyond the convex hull of underlying invariant sets. This benefit is achieved while actually reducing online computational demands.

Guaranteed convergence and recursive feasibility are established for both algorithms for the LPV case. Numerical examples have been used to demonstrate the efficacy of the proposed new GIMPC2 algorithm.

Interesting future research directions are the extension of the new algorithm to LPV systems subject to bounded disturbances and an investigation of the multi-parametric quadratic programming solution for the new algorithm, as this may also give reductions in complexity. The load of GIMPC2 is linked to the state dimension and hence this may restrict its utility on high order systems, so an interesting path is finding interpolation methods that do not need [17] the explicit decomposition of the states implied in (32), but rather can make use solely of the polyhedral invariant sets. This would cause a significant reduction of the number of on-line optimization variables and consequently a reduction of the computational load.

7 Acknowledgments

To the Royal Society and the Royal Academy of Engineering of the United Kingdom.

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. 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 and Dr. Bart De Moor is a full professor, both at the Katholieke Universiteit Leuven, Belgium.

References

[1] Bacic, M. and Cannon, M. and Lee, Y. I. and Kouvaritakis, B., General Interpolation in MPC and Its Advantages, IEEE Transactions on Automatic Control, 48, 6, 2003, 1092–1096

[2] Blanchini, F., Ultimate Boundedness Control for Uncertain Discrete-Time Systems via Set-Induced Lyapunov Functions, IEEE Transactions on Automatic Control, 1994, 39, 428-433

[3] Blanchini, F., Set Invariance in Control, Automatica, 1999, 35, 1747-1767

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

[5] Camacho, E. F., and C. Bordons, Model Predictie Control, 1999, Springer Verlag

[6] Gilbert, E. G. and Tan. K. T., Linear Systems with State and Control Constraints : The Theory and Application of Maximal Output Admissable Sets, IEEE Transactions on Automatic Control, 1991, 36, 9, 1008-1020

[7] Rojas, O.J. and G. C. Goodwin, A simple antiwindup strategy for state constrained linear control, IFAC World Congress, 2002.

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

[9] Kothare, M. V. and Balakrishnan, V. and Morari, M., Robust Constrained Model Predictive Control using Linear Matrix Inequalities, Automatica, 32, 1361-1379, 1996

[10] Lee, Y. I. and Kouvaritakis, B., Robust Receding Horizon Predictive Control For Systems With Uncertain Dynamics and Input Saturation, Automatica, 2000, 36, 1497-1504

[11] Mao, W.-J., 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

[12] Michalska, H. and D. Mayne, Robust receding horizon control of constrained nonlinear systems, IEEE Transactions on Automatic Control, 38, 1623-1633, 1993.

[13] Pluymers, B. and Roobrouck, L. and Buijs, J. and Suykens, J.A.K. and De Moor B., An LMI-based Constrained MPC Scheme with Time-varying Terminal Cost, Proceedings of the IFAC Symposium on Advanced Control Of Chemical Processes, 2004, Hong Kong.

[14] Pluymers, B. and Rossiter, J.A. and Suykens, J.A.K. and De Moor, B., The Efficient Computation of Polyhedral Invariant Sets for Linear Systems with Polytopic Uncertainty, Accepted for publication in Proceedings of the Americal Control Conference, 2005, Portland, USA.

[15] Pluymers, B. and Rossiter, J.A. and Suykens, J.A.K. and De Moor, B., Interpolation Based MPC for LTV Systems using Polyhedral Invariant Sets, Accepted for publication in Proceedings of the Americal Control Conference, 2005, Portland, USA.

[16] Pluymers, B. and Rossiter, J.A. and Suykens, J.A.K. and De Moor, B., Efficient Computation of Polyhedral Invariant Sets for LPV Systems and Application to Robust MPC Submitted for publication, 2005. (http://www.esat.kuleuven.be/ sistawww/cgi-bin/pub.pl)

[17] Pluymers, B. and Roobrouck, L. and Buijs, J. and Suykens, J. A. K. and De Moor, B., Model-Predictive Control with Time-Varying Terminal Cost using Convex Combinations, Automatica 41(5), 831-837, 2005.

[18] Pluymers, B., J.A. Rossiter, J.A.K. Suykens, B. De Moor, A simple algorithm for robust MPC, Accepted for publication in Proceedings of the IFAC World Congress, 2005, Prague, Czech Republic.

[19] Richalet, J., A. Rault, J.L. Testud and J. Papon, Model predictive heuristic control: applications to industrial processes, Automatica, 14(5), 413-428, 1978.

[20] Rossiter, J.A., B. Kouvaritakis and M. Cannon, Computationally efficient algorithms for constraint handling with guaranteed stability and near optimality, 2001, International Journal of Control, 74, 17, 1678-1689

[21] Rossiter, J.A., P.W. Neal and L. Yao, Applying predictive control to fossil fired power station, 2002 Trans. Institute of Measurement and Control, 24, 3, pp177-194

[22] Rossiter, J.A. and Kouvaritakis, B. and Bacic, M., Interpolation based computationally efficient predictive control, International Journal of Control, 77, 3, 290-301, 2004

[23] Rossiter, J.A., Model based predictive control: a practical approach, CRC Press, 2003

[24] Rossiter, J.A., Y. Ding, L. Xi, B. Pluymers, J.A.K. Suykens, B. De Moor, Interpolation based MPC with exact constraint handling: the nominal case, submitted.

[25] Schuurmans, J. and J.A. Rossiter, Robust piecewise linear control for polytopic systems with input constraints, 2000, IEE Proc. Control Theory and Applications, Vol. 147, No. 1, pp13-18

[26] Scokaert, P.O.M. and Rawlings, J.B., Constrained linear quadratic regulation, IEEE Transactions on Automatic Control, 43, 8, 1163-1168, 1998

[27] Tan, K.T. and E.G. Gilbert, Multimode controllers for linear discrete time systems with general state and control constraints, in Optimisation Techniques and Applications (World Scientific, Singapore), 433-442, 1992.

[28] Wan, Z. and Kothare, M. V., Robust output feedback model predictive control using off-line linear matrix inequalities, Journal of Process Control, 12, 763-774, 2002

[29] Wan, Z. and Kothare, M. V., An efficient off-line formulation of robust model predictive control using linear matrix inequalities, Automatica, 39, 5, 837-846, 2003

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