Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Interpolation based robust MPC with exact constraint handling

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

‡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/`
†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/`

***Abstract*— Recent papers (Bacic *et al.*, 2003, Pluymers *et al.*, 2005) have shown that interpolation between different fixed linear control laws can lead to control algorithms with large feasible regions and low on-line computational complexity. The feasible region was shown to be equal to the convex hull of the feasible regions of the different linear control laws between which the interpolation takes place. This paper extends the results of (Pluymers *et al.*, 2005) and shows that the feasible region can be significantly enlarged beyond the convex hull by taking the interaction between the different control laws into account in the calculation of the feasible region. This paper deals with linear systems with polytopic uncertainty description and makes use of polyhedral invariant sets. Rigorous proofs of the results are provided.**

I. Introduction

Model Predictive control (MPC) is an invaluable technique for handling constraints [4], [13], however this can come at the cost of a significant online computational cost. Nevertheless, it is widely accepted that quadratic programming (QP) optimisations are reasonable and hence MPC is heavily used in the process industry. Unfortunately, MPC algorithms leading to a QP optimisation, generally deal with the certain case and it is necessary to assume that either, the inherent robustness of the approach or some form of backoff, will cater for any uncertainty.

Hence, there is much interest ([6]) in how to extend MPC to cater explicitly for parameter uncertainty. This paper focuses on uncertainty modelled as a linear parameter varying (LPV) system. The predominant number of articles in the literature use ellipsoidal invariance as a key tool in establishing the stablity of LPV systems. This is because one can use linear matrix inequalities (LMI) to set up conditions for feasibility, stability and convergence and LMIs give rise to convex optimisations. The flip side however is that the optimisations can be significantly more demanding than a QP and feasible regions are restricted to ellipsoids.

Some authors used ellipsoidal invariance for establishing stability, but posed a simpler variant of MPC to allow for easier optimisations. For instance [7] added degrees of freedom (d.o.f.) through an autonomous model and required only a line search, whereas [1] used General Interpolation (GIMPC) between fixed linear feeback laws. It is the latter of these on which the current paper is based. GIMPC extended feasibility to the convex hull of the invariant ellipsoids allied to several underlying control laws. Crucially, the technique is limited to feasible regions defined via ellipsoids whereas the maximal admissible set (MAS, [5]) is usually polyhedral and often significantly larger [11] than the largest invariant ellipsoid. Hence the paper [1] invited two obvious questions:

1) For the nominal case, can we pose a general interpolation based on polyhedrals rather than ellipsoids and if so how does it compare by way of computational load and feasible region. This topic is discussed in [11] for the nominal case.
2) Can we take the general interpolation based on polyhedrals and apply it to the LPV case? This topic was tackled in [9] and is further extended in the current paper.

Section II will give a quick review of polyhedron based GIMPC for the nominal case and discusses how MAS might be computed for the LPV case [8]. Section III, proposes an extension of [1] to utilise polyhedral, and hence larger volume sets within a GIMPC algorithm [9]. Section IV creates a polyhedral GIMPC algorithm with substantially larger robust feasible regions by taking into account the interaction between the different control laws. Proofs of the properties are included. The paper finishes with examples and conclusions.

II. Background

A. Model and objective

This paper considers LPV systems of the form

$$
\begin{array}{c}
x_{k+1} = A(k)x_k + B(k)u_k, \\
(A(k), B(k)) \in \mathrm{Co}\{(A_1, B_1), ..., (A_m, B_m)\}.
\end{array} \tag{1}
$$

The system is subject to constraints (more general linear constraints can also be considered):

$$
u(k) \in \mathcal{U} \equiv \{u : \underline{u} \le u \le \overline{u}\}, \qquad k = 0, \dots, \infty, \tag{2a}
$$

$$
x(k) \in \mathcal{X} \equiv \{x : \underline{x} \le x \le \overline{x}\}, \qquad k = 0, \dots, \infty. \tag{2b}
$$

$x(k) \in \mathbb{R}^{n_x}$ and $u(k) \in \mathbb{R}^{n_u}$ denote state and input vectors at discrete time k; n_x and n_u are the state and input dimensions respectively.

Assume that one can choose from r different feedbacks $K_j,\ j = 1, ..., r$ (one of these might be the unconstrained optimal, say K_1) with which there are associated closed-loop state matrices:

$$
u = -K_j x,\ \Phi_{ij} = A_i - B_i K_j, \quad j = 1, ..., r,\ i = 1, ..., m. \tag{3}
$$

An underlying objective is, at every sample, to choose a predicted control trajectory (of which only the first is implemented) which minimises the following objective and subject to constraints (2):

$$
J = \sum_{k=0}^{\infty} (x(k)^{\mathrm{T}} Q x(k) + u(k)^{\mathrm{T}} R u(k)), \tag{4}
$$

with $Q = Q^{\mathrm{T}} \in \mathbb{R}^{n_x \times n_x}$ and $R = R^{\mathrm{T}} \in \mathbb{R}^{n_u \times n_u}$ positive definite state and input cost weighting matrices. For the nominal case and optimal control $u = -Kx$, one can express (4) as $J(x) = x^T V_0 x$ (for a suitable $V_0 = V_0^{\mathrm{T}} > 0$).

One requirement of interpolation methods is that there is a quadratic stabilisability condition, that is for any feedback K, there exists a Lyapunov function that applies irrespective of the variation in the process allowed in (1). Hence there must exist $V_j,\ \forall j$ such that:

$$
V_j - \Phi_{ij} V_j \Phi_{ij} \le 0,\ \forall i; \tag{5}
$$

These V_j will not match V_0 in general.

B. Polyhedral Invariant Sets for LPV systems

Under mild conditions, the maximum volume feasible region MAS [5] for a stable linear system with linear constraints is polyhedral. Recently [8] it has been shown that as long as an LPV system is quadratically stabilisable, the same statement holds. For convenience, we give a truncated description of the algorithm to find this set.

1) Assume that an outer approximation to the MAS is given by (2) at $k = 0$ only. Then letting $u(0) = -K_1 x(0)$, this reduces to $\mathcal{S}_o = \{x : M_o x \le d_o\}$ where definitions of $M_o,\ d_o$ are obvious.
2) Set up an iteration on sets $\mathcal{S}_k$ initialised with $\mathcal{S}_o$, such that we find the set $\mathcal{S}_{k-1}$ of previous states $x(-1)$ such that $x \in \mathcal{S}_k$; therefore
$$
S_{k-1} = \{x(0) : x(-1) \in \mathcal{S}_k, \Phi_i x(-1) \in \mathcal{S}_k, \forall i\} \tag{6}
$$
3) Iterate until $\mathcal{S}_{k-1} \equiv \mathcal{S}_k$ and then $\mathcal{S}_k$ is the MAS for the LPV system.

Redundant constraints should be removed regularly or the total number of constraints will explode combinatorially. Let the MAS, be given as $\mathcal{S} = \{x : Mx \le d\}$.

Remark 1: A MAS is invariant, so $x(k) \in \mathcal{S} \ \Rightarrow x(k+i) \in \mathcal{S}, \forall i > 0$, irrespective of the variation of $A(k), B(k)$. Moreover, the trajectories satisfy (2) and, from quadratic stabilisability (5), converge to the origin.

C. General Interpolation (GIMPC): the nominal case [11]

Given a system (1), constraints (2), a set of asymptotically stabilizing feedback controllers (3) and corresponding MAS ($\mathcal{S}^{[j]},\ j = 1, ..., r$), consider the following decomposition:

$$
x(0) = \sum_{j=1}^{r} x_j, \quad \text{with} \quad \left\{ \begin{array}{l} \sum_{j=1}^{r} \lambda_j = 1,\ \lambda_j \ge 0, \\ x_j \in \lambda_j \mathcal{S}^{[j]} \end{array} \right. \tag{7}
$$

This decomposition can be performed iff $x \in \overline{\mathcal{S}}$,

$$
\overline{\mathcal{S}} \triangleq \mathrm{Co}\{\mathcal{S}_1, \dots, \mathcal{S}_r\} \tag{8}
$$

Furthermore given (7) holds [1], the following control law ensures that x remains in $\overline{\mathcal{S}}$:

$$
u(k) = -\sum_{j=1}^{r} K_j x_j, \tag{9}
$$

More generally, define the input and state predictions as:

$$
u(k) = -\sum_{j=1}^{r} K_j \Phi_j^k x_j; \quad x(k) = \sum_{j=1}^{r} \Phi_j^k x_j. \tag{10}
$$

where $\Phi_j = A - BK_j$. For a nominal model, Lyapunov theory can be used to compute the cost (4) as

$$
J = \tilde{x}^{\mathrm{T}} P \tilde{x} = \sum_{k=0}^{\infty} x(k+1)^{\mathrm{T}} Q x(k+1) + u(k)^{\mathrm{T}} R u(k) \tag{11}
$$

$$
\begin{array}{c}
\tilde{x} = [x_1^{\mathrm{T}} \ \dots \ x_n^{\mathrm{T}}]^{\mathrm{T}};\ P \ge \Gamma_u^{\mathrm{T}} R \Gamma_u + \Psi^T \Gamma_x^{\mathrm{T}} Q \Gamma_x \Psi + \Psi^T P \Psi \\
\Psi = [\Phi_1^{\mathrm{T}} \ \dots \ \Phi_r^{\mathrm{T}}]^{\mathrm{T}},\ \Gamma_x = [I \ \dots \ I],\ \Gamma_u = [K_1 \ \dots \ K_r]
\end{array}
$$

Algorithm 1 (GIMPC for the nominal case): Take a system (1), constraints (2), cost weighting matrices Q, R, controllers K_j and invariant sets $\mathcal{S}^{[j]}$ and compute a suitable P from (11). Then, at each time instant, given the current state $x(0)$, solve the following QP optimisation:

$$
\min_{x_j, \lambda_j} \ \tilde{x}^{\mathrm{T}} P \tilde{x}, \quad \text{subject to (7)}, \tag{12}
$$

and implement the input $u = -\sum_{j=1}^{n} K_j x_j$.

Algorithm 1 guarantees recursive feasibility, constraint satisfaction and asymptotic stability. This algorithm can deploy either ellipsoids or polyhedrals for invariant sets $S^{[j]}$; the latter case comprises algorithm 2.1 from [11].

D. Weaknesses of GIMPC [1] and contributions of this paper

1) Feasibility is restricted to $\overline{\mathcal{S}}$ (8).
2) Polyhedral algorithm is currently only applicable to the nominal case.
3) Algorithm for LPV case uses ellipsoids and hence has restricted feasibility.

This paper proposes an algorithm overcoming all three weaknesses. First we show how to extend the use of polyhedral MAS to the LPV case within the context of GIMPC (along the same lines as [9]) and secondly how to improve constraint handling to improve feasibility beyond the convex hull of (8).

III. Polyhedron based GIMPC for the uncertain case

This brief section extends the GIMPC algorithm to make use of polyhedral sets in the uncertain case. In summary, take the cost function given in [1] but replace the ellipsoidal invariant sets with those defined in [8].

Definition 1 (Cost function for LPV case): Take the cost function defined in (11,11) for the nominal case. Replace Ψ by $\Psi_i = [\Phi_{i1} \ \ \cdots \ \ \Phi_{ir}],\ i = 1, ..., m$ and compute a P that is the least upper bound for each of these Ψ_i.

Definition 2 (Invariant sets): Take the algorithm summarised in section II-B and, independently for each K_i, find the robust MAS $\mathcal{S}^{[i]} = \{x : M_i x \le d_i\}$.

Algorithm 2: [Polyhedral GIMPC for the LPV case]

1) Define the sets $\mathcal{S}^{[j]},\ j = 1, ..., r$ for the r different feedbacks $K_j,\ j = 1, ..., r$ corresponding to the LPV system/constraints (1,2).
2) Define an appropriate least upper bound $J = \tilde{x}^{\mathrm{T}} P \tilde{x}$ for LPV system made up from (3).
3) Use sets $\mathcal{S}^{[j]}$ and cost J in the algorithm 1.

Theorem 1: Algorithm 2 has a guarantee of recursive feasibility and a gaurantee of convergence when a applied to system (1) [1].

Proof: Decomposition (7) ensures that feasibility now implies feasibility at the next step and for the entire implied prediction . Also, by definition, for any valid choice of $\lambda_j,\ x_j,\ J$ is Lyapunov and hence one can be sure that the state converges to the origin. $\square$

IV. Increasing feasible regions for polyhedral GIMPC in the uncertain case

In this section we propose how to extend the feasible region of GIMPC to a region larger than the convex hull of the feasible regions of the different. We will call the new algorithm GIMPC2. GIMPC2 contains as a subset all solutions available to GIMPC and yet achieves this with fewer d.o.f. and while giving larger feasible regions.

A. Extension of GIMPC to GIMPC2

First, we briefly discuss some insights regarding the constraint handling philosophies of GIMPC and GIMPC2, and hence show how one might extend GIMPC in order to remove conservative constraint handling.

Let the MAS for system (1) under feedback K_i be:

$$\mathcal{S}^{[j]} = \{x : M_j x \le d_j\} \tag{13}$$

1) Constraints for GIMPC: The GIMPC constraints of (7) can also be posed as:

$$\begin{bmatrix} M_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_r \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_r \end{bmatrix} \le \begin{bmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_r \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_r \end{bmatrix} \tag{14}$$

with $\sum_j \lambda_j = 1,\ \ \lambda_j \ge 0,\ \ \sum_j x_j = x$. It is easy to see that this does implicit constraint handling in that it adds worst case maxima and minima associated to each component x_j without any regard to whether these peaks occur at the same sampling instant. As a consequence, this approach is conservative and one could easily find scenarios where the closed-loop trajectories never come near to a constraint.

2) Constraints for GIMPC2: GIMPC2 relies on the MAS of (13) having a particular structure (GIMPC does not). Specifically, let the inequalities defined by the kth row of $M_j,\ d_j$ correspond to a particular constraint (for instance the j-step ahead prediction of the input being on an upper constraint). Then, the kth row of $M_l,\ d_l,\ \ \forall l \ne i$ must also correspond to the same constraint. As a consequence, $M_j,\ d_j,\ \forall j$ must have the same total number of rows.

GIMPC2 then does explicit constraint handling in that it adds the predictions associated to each component x_i and checks the total prediction against constraints. This operation can be summarised in the constraints:

$$\begin{bmatrix} M_1 & \cdots & M_r \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_r \end{bmatrix} \le \begin{bmatrix} d_1 & \cdots & d_r \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_r \end{bmatrix} \tag{15}$$

with $\sum_j \lambda_j = 1,\ \ \sum_j x_j = x$. These are clearly simpler than (14) and moreover can be made simpler still if one realises that the λ_j variables are now superflous. First normalise the inequalities in (13) so that $d_1 = d_2 = ... = d_r = d$, then given the condition $\sum_j \lambda_j = 1$, the right hand side of (15) reduces to just d, i.e.:

$$\begin{bmatrix} M_1 & \cdots & M_r \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_r \end{bmatrix} \le d; \quad \sum_j x_j = x \tag{16}$$

Remark 2: The reader will also note that, unlike GIMPC, there is no need for the condition $\lambda_j \ge 0$. Values for λ_i could be implied but are not needed.

Remark 3: Even though the different M_i define invariant sets for the individual Φ_i, it is not guaranteed that (16) defines an invariant set for the total system with state vector $[x_1^{\mathrm{T}} \;\; \dots \;\; x_r^{\mathrm{T}}]$. However, in order to obtain a recursively feasible algorithm, invariance of the constraint set is required. Therefore, in the next section, the constraints are calculated as the invariant set of one of two possible augmented systems.

B. Comparison of GIMPC2 over GIMPC

This section gives a brief review, including the pros and cons, of these two approaches.

- GIMPC was originally developed for ellipsoidal sets and used implicit constraint handling (7), hence giving reduced feasibility regions. This weakness carried over to the polyhedral implementation.
- GIMPC2 introduces explicit constraint handling to the interpolation and hence has larger feasible regions, despite using the same underlying sets $\mathcal{S}^{[j]}$.
- GIMPC2 requires fewer d.o.f. than GIMPC because it does not require the λ_i variables.
- In GIMPC M_j, d_j can be reduced to minimal form. GIMPC2 must include every constraint required to define the MAS for any of the K_j; hence the set definitions (13) may require more rows.
- GIMPC extends easily to the LPV case. This is not the case for GIMPC2, because of the need to impose mutually consistent structures for $\mathcal{S}^{[j]}$. Algorithms to do this are developed next.

C. Constraint calculation for GIMPC2

GIMPC2 uses fewer variables and has wider feasibility than GIMPC, however the constraints have to be formulated such that invariance for the total system is obtained (enabling a recursive feasibility proof), while still using exact constraint handling. For this reason the constraints cannot be constructed based on the MAS for the different controllers, but have to be constructed as the MAS for an augmented system.

1) Method 1: We first construct an augmented system (i.e., a system with increased dimensionality) and then use the standard algorithm of [8] to deal with the constructed LPV model (1).

Given control law (9) and state decomposition:

$$x = \sum x_i \quad \Rightarrow \quad x_r = x - x_1 - x_2 - \dots - x_{r-1} \tag{17}$$

define an augmented state

$$X = \begin{bmatrix} x \\ x_1 \\ \vdots \\ x_{r-1} \end{bmatrix} \tag{18}$$

and hence an augmented LPV controlled system as $X(k+1) = \Psi(k)X(k)$, $\Psi(k) \in Co\{\Psi_1, \dots, \Psi_m\}$

$$\Psi_i = \begin{bmatrix} A_i - B_iK_r & B_i(K_r - K_1) & \cdots & B_i(K_r - K_{r-1}) \\ 0 & A_i - B_iK_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_i - B_iK_{r-1} \end{bmatrix} \tag{19}$$

Constraints (2) should be written in terms of X and then, using the algorithm of section II-B on the augmented system, the MAS will be of the form

$$\mathcal{S}_a = \{X : M_aX \leq d_a\} \tag{20}$$

The projection to $x-$space can be defined as

$$\mathcal{S}_{ax} = \{x : \exists X \text{ s.t. } M_aX \leq d_a\} \tag{21}$$

or rearranged into the form of (15).

2) Method 2: Alternatively one can construct a different state vector as

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_r \end{bmatrix}, \tag{22}$$

and a corresponding augmented controlled LPV system as

$$\Psi_i = \begin{bmatrix} A_i - B_iK_1 & 0 & \cdots & 0 \\ 0 & A_i - B_iK_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_i - B_iK_r \end{bmatrix}. \tag{23}$$

The MAS for this set can then again be used as a constraint set in the GIMPC2 algorithm. We note that it is important to set up the constraints to ensure that the relation $x = \sum_i x_i$ is implied in all the inequalities.

D. Feasibility and stability

It can be easily shown that 14 (after elimination of the λ_i) is an invariant set for the augmented LPV system (23) and hence a subset of the MAS of this system. Therefore the feasibility region of GIMPC is also a subset of the feasibility region of GIMPC2.

1) GIMPC2 has recursive feasibility: The feasible region $\mathcal{S}_a$ is constructed on the basis of invariance, that is, all possible future predictions for X remain within the region $\mathcal{S}_a$. It then follows automatically that $X \in \mathcal{S}_a \quad \Rightarrow \exists(x_i, i = 1, \dots, r)$ s.t. $x \in \mathcal{S}_{ax}$.

2) Convergence: All that remains therefore is to establish the convergence of the GIMPC2 algorithm with constraints (20) when applied to system (1). For this, the reader is referred back to Theorem 1. One can establish gauranteed convergence if the optimisation cost J is replaced by an appropriate upper bound $\tilde{x}P\tilde{x}$ such that for system (1) and control laws K_i, one can be sure that there exist $\tilde{x}$ such that J is monotonically decreasing and therefore Lyapunov. A suitable upper bound is given in [1].

V. EXAMPLES

Examples are used to demonstrated the potentially large increase in feasibility obtained by using GIMPC2 in place of GIMPC. Also, some closed-loop simulations will illustrate the efficacy of the algorithm for controlling the uncertain system. Finally, for completeness, some discussion is given to the relative complexity of the computations.

A. Feasibility regions

Consider the LPV system and constraints:

$$\begin{array}{c} A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}; B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \\ A_2 = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix}; B_2 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}; \end{array} \tag{24}$$

$$\overline{u} = 1, \qquad \underline{u} = -1, \tag{25}$$

$$\overline{x} = [10, \ 10]^{\mathrm{T}} \qquad \underline{x} = [-10, \ -10]^{\mathrm{T}}. \tag{26}$$

Two robustly stablising control laws are

$$K_1 = [-0.3 \ -0.1]; \quad K_2 = [-0.5 \ -0.3] \tag{27}$$

Figure 1 plots the associated MAS ($\mathcal{S}_1$, $\mathcal{S}_2$) (ellipsoidal in dashed line and polytopic in solid line). The convex hull (8) (Feasible region for GIMPC) and the feasible region for GIMPC2 $\mathcal{S}_{ax}$ are shown in figure 2. Finally, figure 3 overlays the feasible regions for a robust[1] optimal control law with 1-5 d.o.f. (denoted as n_c). Several things are clear:

- We have successfully combined robust polyhedral MAS and hence allowed explicit constraint handling for the LPV case.
- GIMPC2 has a larger feasible region than GIMPC.
- With just 2 d.o.f (in this example), GIMPC has far better feasibility than conventional robust MPC algorithms [14] using similar numbers of d.o.f.

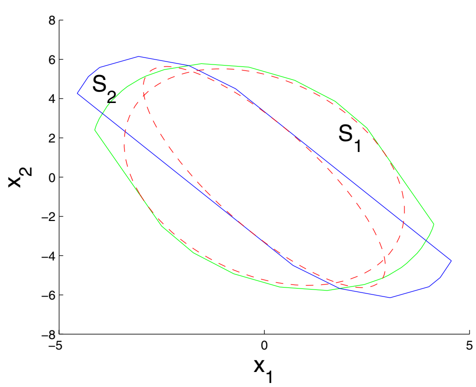

Fig. 1. Invariant ellipsoids and polyhedrals for linear feedbacks.

[1]A robust version of this using robust polyhedral invariant sets is discussed in [14].

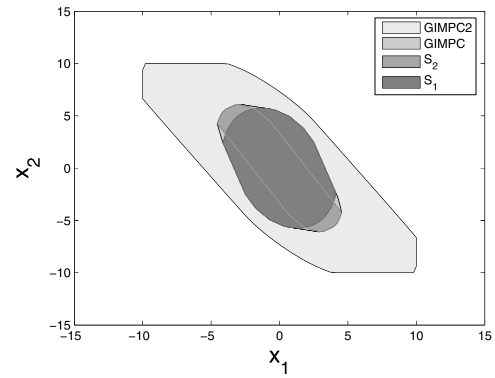

Fig. 2. Invariant polyhedrals, the convex hull and S_{ax}.

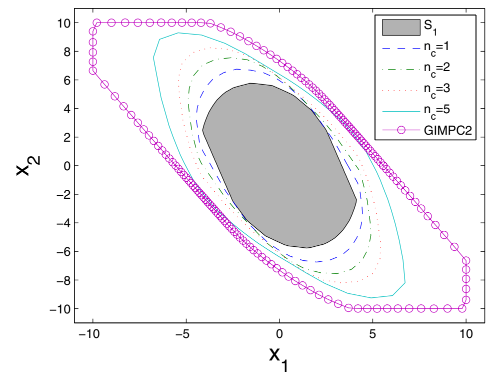

Fig. 3. Invariant polyhedrals for robust MPC and S_{ax}.

B. Closed-loop simulations

Next, we demonstrate that GIMPC2 gives convergent behaviour from all points within $\mathcal{S}_{ax}$. Figure 4 shows the state trajectories for several initial points on the boundary. For each trajectory, $A(k)$, $B(k)$ are deterministic, but sampled randomly within the limits of (1), yet all the trajectories remain within $\mathcal{S}_{ax}$ and moreover converge to the origin; earlier papers have shown that a failure to use robust invariant sets will often lead to divergent behaviour [14]. It is also worth noting that these trajectories show a distinctive time varying nature especially when the state nears $\mathcal{S}_1$ (marked in figure). So despite deploying so few d.o.f., the control law has embedded a large degree of flexibility.

C. Computational issues

Two comparisons are in order: (i) the number of variables required in the QP optimisation and (ii) the dimensions of the inequalities in the QP optimisation. One could also make this comment in relation to algorithms which use ellipsoidal invariant sets but the optimisations there are more demanding [1]. It is evident that for this example:

- GIMPC2 use less d.o.f. then both GIMPC and RMPC, despite having a larger feasible region.

- GIMPC2 and RMPC both require substantial more inequalities than GIMPC.

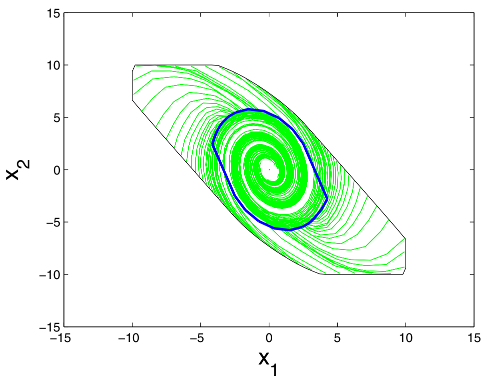

Fig. 4. Closed-loop trajectories for GIMPC2.

GIMPC	GIMPC2	Robust MPC
$(r-1)(n_x+1)=3$	$(r-1)n_x=2$	n_c

Table 1: Number of variables in optimisation

	M_1	M_2	other	Total
Rows (GIMPC)	30	12	2	44
Rows (GIMPC2)	412	412	0	412
Rows (Robust MPC $n_c=5$)				448

Table 2: Number of inequalities in optimisation

VI. Conclusions

This paper shows that GIMPC using polyhedral invariant sets for LPV systems can exhibit conservative constraint handling and proposes a new algorithm that is able to significantly extend the feasible region beyond the convex hull of the feasible regions of the different control laws. Necessary conditions are discussed and suitable modifications to the robust MAS algorithm of [8] are proposed and implemented.

A numerical example demonstrates the efficacy of the new interpolation. Simulation studies demonstrate the low computational load and yet large feasible regions of the proposed interpolation algorithm. The main drawback is the large number of inequalities; current work is considering ways of reducing this.

VII. ACKNOWLEDGMENTS

Thanks to the Royal Society & Royal Academy of Engineering. Research partially supported by Research Council KULeuven: GOA-Mefisto 666, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects, G.0240.99, G.0407.02, G.0197.02, G.0141.03, G.0491.03, G.0120.03, research communities (ICCoS, ANMMM); AWI: Bil. Int. Collaboration Hungary/ Poland; IWT: PhD Grants, Soft4s (softsensors), Belgian Federal Government: DWTC (IUAP IV-02 (1996-2001) & 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. B.Pluymers is a Research Assistant with the I.W.T. (Flemish Institute for Scientific and Technological Research in Industry), Johan Suykens is an associate professor and Bart De Moor a full professor, all 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, June, 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] Camacho, E. and C. Bordons, Model predictive Control, Springer Verlag, 1999

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

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

[7] Kouvaritakis, B., J.A. Rossiter and J. Schuurmans, 2000, Efficient robust predictive control, IEEE Trans AC, 45, 8, pp1545-1549

[8] 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, Proceedings of the ACC'05, Portland, USA, 2005

[9] Pluymers, B. and Rossiter, J.A. and Suykens, J.A.K. and De Moor, B., Interpolation Based MPC for LPV Systems using Polyhedral Invariant Sets, Proceedings of the ACC'05, Portland, USA, 2005

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

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

[12] J.A. Rossiter, P.W. Neal and L. Yao, Applying predictive control to fossil fired power station, Proc. Inst. MC, 24(3), 177-194, 2002.

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

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

[15] Scokaert, P.O.M. and Rawlings, J.B., Constrained linear quadratic regulation, IEEE Trans. AC, 43, 8, 1163-1168

[16] 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, Internal Report 04-28, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), to appear in Automatica, 2005