2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Interpolation Based MPC for LPV Systems using Polyhedral Invariant Sets

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/

Abstract— **Guaranteeing asymptotic stability and recursive constraint satisfaction for a set of initial states that is as large as possible and with both a minimal control cost and computational load can be identified as a common objective in the Model Predictive Control (MPC) community. General interpolation (Rossiter *et al.*, 2004, Bacic *et al.*, 2003) provides a favourable trade off between these different aspects, however, in the robust case, this requires on-line Semi-Definite Programming (SDP), since one typically employs ellipsoidal invariant sets. Recently, (Pluymers *et al.*, 2005) have proposed an efficient algorithm for constructing the robust polyhedral maximal admissible set (Gilbert *et al.*, 1991) for linear systems with polytopic model uncertainty. In this paper a robust interpolation based MPC method is proposed that makes use of these sets. The algorithm is formulated as a Quadratic Program (QP) and is shown to have improved feasibility properties, efficiently cope with non-symmetrical constraints and give better control performance than existing interpolation based robust MPC algorithms.**

I. INTRODUCTION

Model based Predictive Control (MPC) classically considers a finite future time horizon within which the input sequence is parameterized with a finite number of degrees of freedom (d.o.f.), after which an optimization problem is solved at each time instant in order to minimize a performance objective, that is typically a quadratic function with Q and R as state and input weighting matrices. Stability and satisfactory performance is generally achieved by considering sufficiently large horizons. Later results (e.g. [13]) achieve stability with more modest horizon lengths by adding a stabilizing feedback controller at the end of the horizon. The terminal state is then restricted to lie within the invariant set of the terminal controller and the cost objective is augmented with the total cost-to-go beyond the horizon. The within-horizon part of the controller is typically called *mode 1*, while the part beyond the horizon is generally called *mode 2*.

A typical disadvantage of *dual mode* MPC controllers is the trade-off to be made between satisfactory control performance (by using a highly-tuned terminal controller) and feasibility (by using a detuned terminal controller with a large region of attraction). Both optimal performance and satisfactory feasibility can be obtained by employing a time-varying terminal controller as proposed in e.g. [1], [4], [9], [10], [16], but this comes at the cost of a significant increase in computational complexity due to the necessity to use Semi-Definite Programming (SDP).

Given that making the terminal controller time-varying is often more effective for enlarging the set of feasible initial states than adding additional d.o.f. in *mode 1*, several *mode 2* only algorithms have been proposed in literature, e.g. [1], [7], [12], [14], [15]. These algorithms can be seperated in two groups. The first family [14], [15] is basically based on the well known results of [7] and use the convexity property of LMI's and several off-line calculated controllers and corresponding ellipsoidal invariant sets to on-line calculate a time-varying controller. The second family, usually called *general interpolation* [1], [12], uses an on-line decomposition of the current state, with each component lying in a seperate invariant set, after which the corresponding controller is applied to each component seperately in order to calculate an input value.

The method proposed in this paper fits in the second category of *mode 2* only algorithms. While [12] is based on QP and is computationally efficient, it cannot take model uncertainty into account whereas the algorithm described in [1] can accommodate for model uncertainty but makes use of more conservative ellipsoidal invariant sets, as opposed to polyhedral sets in [12]. Hence, it can only be formulated as an SDP, which makes the method computationally more demanding. This paper uses the algorithm recently proposed in [11] for constructing polyhedral robustly invariant sets (previously considered intractable), in order to obtain a QP based robust interpolation based algorithm with improved control performance and improved feasibility properties.

This paper is organised as follows. In section II the general control setting and notation is introduced, after which, in

0-7803-9098-9/05/$25.00 ©2005 AACC

section III, the general background on invariant sets and general interpolation are briefly explained. Section IV then introduces the new algorithm, which is then applied to a numerical example in section V. Sections VI and VII contain conclusions and future work.

II. PROBLEM FORMULATION

In this paper we consider linear parameter-varying (LPV) systems of the form

$$x(k+1) = A(k)x(k)+B(k)u(k), \qquad k = 0,\dots,\infty \quad (1a)$$

with

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

subject to constraints

$$u(k) \in \mathcal{U} \equiv \{u|\underline{u} \leq u \leq \overline{u}\}, \qquad k = 0,\dots,\infty, \quad (2a)$$

$$x(k) \in \mathcal{X} \equiv \{x|\underline{x} \leq x \leq \overline{x}\}, \qquad k = 0,\dots,\infty. \quad (2b)$$

$x(k) \in \mathbb{R}^{n_x}$ and $u(k) \in \mathbb{R}^{n_u}$ denote state and input vectors at discrete time k with n_x and n_u respectively denoting the number of states and inputs of the system. For reasons of clarity component-wise bounds are assumed, but more general linear state, input and mixed state/input constraints can also be considered without significantly complicating further sections. In this paper a new algorithm is proposed that robustly stabilizes system (1) and guarantees robust satisfaction of constraints (2). The algorithm aims to minimise $\sum_{k=0}^{\infty}(x(k)^{\mathrm{T}}Qx(k)+u(k)^{\mathrm{T}}Ru(k))$ as a cost objective with $Q \in \mathbb{R}^{n_x\times n_x}$ and $R \in \mathbb{R}^{n_u\times n_u}$ positive definite state and input cost weighting matrices.

III. BACKGROUND

In this section an overview is given of the existing methods for computing invariant sets and how these are deployed in MPC algorithms using general interpolation. The next section will then focus on the new algorithm proposed in this paper.

A. Invariant Sets

As this section only gives a brief overview of invariant sets and how to construct them, focusing on the details required to understand general interpolation, we refer the reader to [3] for further details.

Definition 1 (Feasibility): Given a dynamical system (1), an asymptotically stabilizing feedback controller $u(k) = -Kx(k)$ and constraints (2), then a set $\mathcal{S} \subset \mathbb{R}^{n_x}$ is feasible iff $\mathcal{S} \subset \{x|x \in \mathcal{X}, -Kx \in \mathcal{U}\}$.

Definition 2 (Robust Positive Invariance): Given a dynamical system (1), a stabilizing feedback controller $u(k) = -Kx(k)$ and constraints (2), then a set $\mathcal{S} \subset \mathbb{R}^{n_x}$ is robust positive invariant iff

$$x(k) \in \mathcal{S} \quad \Rightarrow \quad (A(k) - B(k)K)x(k) \in \mathcal{S}, \\ \forall [A(k)\ B(k)] \in \Omega,\ k = 0,\dots,\infty. \quad (3)$$

The largest possible feasible invariant set is generally called the Maximal Admissable Set (MAS, [5]). The MAS for an LTI system ((1) with $m = 1$) is given by $\mathcal{S} = \bigcap_{k=0}^{\infty}\{x|(A-BK)^k x \in \mathcal{X}, -K(A-BK)^k x \in \mathcal{U}\}$. Under certain conditions on the feedback controller K this set can be shown [5] to be equal to $\mathcal{S} = \bigcap_{k=0}^{n}\{x|(A-BK)^k x \in \mathcal{X}, -K(A-BK)^k x \in \mathcal{U}\}$ with n a finite number, indicating that $\mathcal{S}$ can be described by a finite number of linear inequalities and that $\mathcal{S}$ therefore is polyhedral.

The extension of the MAS to LPV systems is theoretically relatively straightforward, but practically not directly applicable since all possible future predictions have to be taken into account, which causes a combinatorial increase in the number of constraints describing $\mathcal{S}$. Therefore, for LPV systems invariant sets of ellipsoidal form $\mathcal{S} = \{x|x^{\mathrm{T}}Z^{-1}x \leq 1\}$ are typically used, since these can be calculated using results from [7], by solving an SDP. However, these are inner approximations to the real MAS and are hence conservative and furthermore, they cannot cope efficiently with non-symmetrical constraints. However, new developments [11] now allow the construction of polyhedral invariant sets for LPV systems and these underpin the algorithm proposed in this paper. Section IV-A contains a summary if these developments.

B. General Interpolation

Given a system (1), constraints (2), a set of asymptotically stabilizing feedback controllers $u(k) = -K_i x(k), i = 1,\dots,n$ and corresponding invariant sets $\mathcal{S}_i, i = 1,\dots,n$, the following decomposition is performed:

$$x = \sum_{i=1}^{n}\hat{x}_i, \quad \text{with} \quad \begin{cases} \sum_{i=1}^{n}\lambda_i = 1,\ \lambda_i \geq 0, \\ x_i \in \mathcal{S}_i, \\ \hat{x}_i = \lambda_i x_i. \end{cases} \quad (4)$$

This decomposition can be performed iff $x \in \overline{\mathcal{S}} \triangleq \mathrm{Co}\{\mathcal{S}_1,\dots,\mathcal{S}_n\}$. The following control law

$$u(k) = -\sum_{i=1}^{n}K_i\prod_{j=0}^{k-1}\Phi_i(k-1-j)\hat{x}_i, \quad (5)$$

with $\Phi_i(k) = A(k) - B(k)K_i$, can be proven to keep the state within $\overline{\mathcal{S}}$. The corresponding state sequence can be calculated to be

$$x(k) = \sum_{i=1}^{n}\prod_{j=0}^{k-1}\Phi_i(k-1-j)\hat{x}_i. \quad (6)$$

One can easily prove that (5) and (6) satisfy input and state constraints respectively. Moreover, one can apply Lyapunov theory in a straightforward manner to compute the infinite-horizon cost $\sum_{k=0}^{\infty}x(k+1)^{\mathrm{T}}Qx(k+1) + u(k)^{\mathrm{T}}Ru(k)$ corresponding to (5)-(6) as the quadratic Lyapunov function $\tilde{x}^{\mathrm{T}}P\tilde{x}$, with $\tilde{x} = [\hat{x}_1^{\mathrm{T}}\ \dots\ \hat{x}_n^{\mathrm{T}}]^{\mathrm{T}}$ and

$$P \geq \Gamma_u^{\mathrm{T}}R\Gamma_u + \Psi_i^T\Gamma_x^{\mathrm{T}}Q\Gamma_x\Psi_i + \Psi_I^T P\Psi_i, \quad i = 1,\dots,n, \quad (7)$$

where $\Psi_i = \text{diag}((A_i - B_iK_1), \ldots, (A_i - B_iK_n))$, $\Gamma_x = [I, \ldots, I]$, $\Gamma_u = [K_1, \ldots, K_n]$. One can calculate P by solving the SDP

$$\min_{P} \ \text{tr}(P), \quad \text{subject to (7).} \tag{8}$$

An MPC algorithm can now be formulated as follows :

Algorithm 1 (MPC using general interpolation): *Given a system* (1)*, constraints* (2)*, cost weighting matrices* Q, R*, controllers* K_i *and invariant sets* $\mathcal{S}_i$*, perform the following steps. First calculate* P *by solving* (8)*. Then, at each time instant, given the current state* x*, solve the following optimisation problem on-line*

$$\min_{\hat{x}_i, \lambda_i} \ \tilde{x}^{\mathrm{T}} P \tilde{x}, \quad \textit{subject to } (4), \tag{9}$$

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

Algorithm 1 guarantees recursive feasibility, constraint satisfaction and asymptotic stability and comprises the mode 2 only algorithm from [1] that uses ellipsoidal invariant sets for LPV systems as well as algorithm 2.1 from [12] that uses polyhedral invariant sets for LTI systems. Detailed stability and feasibility proofs can be found in both references. In the sequel we refer to these algorithms as REMPC and IMPC respectively.

IV. GENERAL INTERPOLATION FOR LPV SYSTEMS USING POLYHEDRAL INVARIANT SETS

A. Polyhedral Invariant Sets For LPV Systems

This paper proposes to make use of a recent development [11] which demonstrates how, in some cases, one can indeed formulate the MAS for an LPV system. The key idea used is not dissimilar to the one-step sets popularised in [6], that is to use backwards prediction rather than forwards prediction. This simple change eliminates the combinatorial explosion in the possible number of prediction terms and hence creates a tractable problem. A brief summary of the key results is given next without details, for which the reader is referred to [11]. First define the closed-loop system matrices

$$\Phi_i = A_i - B_i K, \qquad i = 1, \ldots, m. \tag{10}$$

The MAS for the uncertain system (1), constraints (2) and control law $u = -Kx$ is:

$$\mathcal{S} = \{x | Mx \leq d\}. \tag{11}$$

By definition $\mathcal{S}$ is invariant so $x \in \mathcal{S} \ \Rightarrow \Phi_i x \in \mathcal{S}$, $i = 1, \ldots, m$. This can be shown to be equivalent to $\mathcal{S} \subset \mathcal{S}^-$, with the $^-$-operator defined as $\mathcal{S}^- = \{x | \Phi_i x \in \mathcal{S}, i = 1, \ldots, m\}$. The following algorithm starts with the initial set $\mathcal{S} = \{x | \underline{x} \leq x \leq \overline{x}, \underline{u} \leq -Kx \leq \overline{u}\}$ and iteratively adds constraints from $\mathcal{S}^-$ until $\mathcal{S} \subset \mathcal{S}^-$. The resulting set is the MAS.

Algorithm 2 (Robust Invariant Set):

1) *Set* $M := [I^{\mathrm{T}} \ -I^{\mathrm{T}} \ -K^{\mathrm{T}} \ K^{\mathrm{T}}]^{\mathrm{T}}, d := [\overline{x}^{\mathrm{T}} \ \underline{x}^{\mathrm{T}} \ \overline{u}^{\mathrm{T}} \ \underline{u}^{\mathrm{T}}]$ *and* $i := 1$.
2) *Select row* i *from* (M, d) *and check* $\forall j$ *whether* $M_i \Phi_j x \leq d_i$ *is redundant with respect to the constraints defined by* (M, d)*. Add the non-redundant constraints to* (M, d) *by assigning* $M := [M^{\mathrm{T}} \ (M_i \Phi_j)^{\mathrm{T}}]^{\mathrm{T}}$ *and* $d := [d^{\mathrm{T}} \ d_i^{\mathrm{T}}]^{\mathrm{T}}$ *for all relevant* j*.*
3) *Set* $i := i + 1$*. If* i *is strictly larger than the number of rows in* (M, d) *then terminate, otherwise continue with step 2).*

The resulting set $\mathcal{S} = \{x | M_u x \leq d_u\}$ is the MAS for the given system, constraints, and feedback controller. The algorithm is guaranteed to terminate in a finite number of iterations if the closed-loop system is quadratically stable :

$$\exists V = V^{\mathrm{T}} > 0 \ \text{ s.t. } \ \Phi_i^{\mathrm{T}} V \Phi_i \leq V, \quad i = 1, \ldots, m. \tag{12}$$

Remark 1: Constraints added in later iterations of the algorithm can render constraints added in earlier iterations redundant. Therefore it is advisable to check for redundant constraints regularly during the execution of the algorithm; this can decrease execution time considerably.

B. Interpolation based MPC for LPV systems using polyhedral invariant sets

The polyhedral invariant sets constructed using algorithm 2 can now be used to formulate a new interpolation based MPC algorithm for LPV systems similar to REMPC, but that makes use of QP instead of SDP; it also fits in the general framework of algorithm 1.

Algorithm 3 (RPMPC): *Given a system* (1)*, constraints* (2)*, cost weighting matrices* Q, R*, asymptotically stabilizing controllers* K_i*, corresponding polyhedral robust invariant sets* $\mathcal{S}_i = \{x | M_i x \leq d_i\}$ *and* P *satisfying* (7)*, solve on-line at each time instant, given the current state* x*, the following problem :*

$$\min_{\hat{x}_i, \lambda_i} \ \tilde{x}^{\mathrm{T}} P \tilde{x}, \tag{13a}$$

$$\textit{subject to} \quad x = \sum_{i=1}^{n} \hat{x}_i, \tag{13b}$$

$$M_i \hat{x} \leq \lambda_i d_i, \qquad i = 1, \ldots, n, \tag{13c}$$

$$\sum_{i=1}^{n} \lambda_i = 1, \tag{13d}$$

$$\lambda_i \geq 0, \qquad i = 1, \ldots, n, \tag{13e}$$

and implement input $u = -\sum_{i=1}^{n} K_i \hat{x}_i$*.*

Lemma 1: Algorithm 3 guarantees robust satisfaction of (2) and is recursively feasible and asymptoticaly stable for all initial states $x(0) \in \text{Co}\{\mathcal{S}_1, \ldots, \mathcal{S}_n\}$.

Proof: It is clear from (4) that (13) is feasible for all $x \in \overline{\mathcal{S}}$. Given the current state $x(k)$, components $\hat{x}_i(k)$ and factors $\lambda_i(k)$, it is possible to calculate the next state to be $x(k) = \sum_{i=1}^{n} \Phi_i(k)\hat{x}_i(k)$. Since the components $x_i(k)$ lie in their respective invariant sets $\mathcal{S}_i$, this will also be the case for the components $x_i(k+1) = \Phi_i(k)x_i(k)$, which shows that $x(k+1)$ will also lie within $\overline{\mathcal{S}}$. By recursively applying this argument it is proven that $x(k+i) \in \overline{\mathcal{S}}, i = 1, \ldots, \infty$. Since all $\mathcal{S}_i$ are subsets of $\mathcal{X}$ and $\mathcal{X}$ is convex, $\overline{\mathcal{S}}$ will also be a subset of $\mathcal{X}$, which then proves robust satisfaction of the state constraints. Furthermore, since $\hat{x}_i \in \lambda_i \mathcal{S}_i$, it is clear that $\hat{u}_i \triangleq -K_i\hat{x}_i \in \lambda_i\mathcal{U}$. Therefore (since $\mathcal{U}$ is a convex set) $u \triangleq \sum_{i=1}^{n} \hat{u}_i \in \mathcal{U}$, which proves robust satisfaction of the input constraints.

Asymptotic stability can be proven by considering components $\hat{x}_i(k)$, which are shown above to provide a feasible candidate decomposition $\Phi_i(k)\hat{x}_i$ for time $k+1$. It is now easy to see, based on satisfaction of (7) that this candidate decomposition already provides a lower value of the cost function $\tilde{x}^{\mathrm{T}}P\tilde{x}$ than the optimal cost value at time k, which proves that the optimal value of the cost at time $k+1$ will also be lower than the optimal value at time k. This consequently proves that the optimal value of the cost function of (13) acts as a Lyapunov function of the closed-loop system, which proves asymptotic stability. ■

In the sequel we refer to algorithm 3 as RPMPC. The algorithm has several advantages compared to REMPC. First of all, it has an enlarged feasibility region, since the individual invariant sets are larger than their ellipsoidal counterparts used in REMPC. Furthermore, the polyhedral invariant sets can efficiently cope with non-symmetrical constraints, which ellipsoids cannot. Due to the invariant sets being larger, less conservative satisfaction of the imposed input and state constraints can be expected, potentially leading to a reduction in control cost. Finally, the algorithm is formulated as a QP, which is significantly less expensive to solve than the SDP formulation of REMPC. These advantages are clearly illustrated in the next section.

V. NUMERICAL EXAMPLE

In this section we use a numerical example to demonstrate the properties of the new algorithm. For simplicity and for ease of comparison with results reported in the literature we will use the double integrator, for this purpose extended with model uncertainty. The model and constraints are given by :

$$A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad (14)$$

$$A_2 = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \qquad (15)$$

$$\overline{u} = 1, \qquad \underline{u} = -0.5, \qquad (16)$$

$$\overline{x} = [8,\ 8]^{\mathrm{T}} \qquad \underline{x} = [-10,\ -10]^{\mathrm{T}}. \qquad (17)$$

The constraints are chosen to be non-symmetrical, in order to demonstrate the ability of the algorithm to efficiently cope with this setting. To facilitate comparison with nominal MPC schemes, we also consider a nominal model chosen as $A = 0.5(A_1 + A_2), B = 0.5(B_1 + B_2)$. Two feedback controllers are chosen; one as the LQR-optimal controller $K = [0.4558\ 0.3698]^{\mathrm{T}}$ for the nominal system and $Q = \mathrm{diag}(1, 0.01), R = 3$; the second as a detuned controller $K = [0.1\ 0.5]^{\mathrm{T}}$. Both controllers are robustly asymptotically stabilizing.

A. Feasible Region

The feasible regions of the algorithms under consideration are equal to the convex hulls of the invariant sets $\mathcal{S}_i$ and hence the invariant set of RPMPC is expected to be significantly larger than REMPC. This difference can clearly be observed in figure 1 where the darker shaded region represents REMPC and the lighter shaded region is for RPMPC; the underlying invariant ellipsoids/polyhedra are marked with solid lines. For completeness, the impact

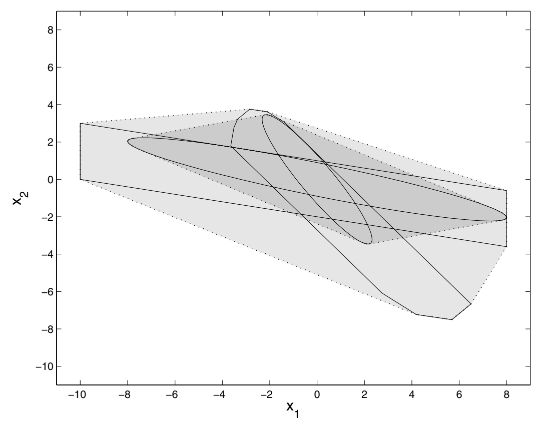

Fig. 1. Feasible region comparison of REMPC (dark shading) and RPMPC (light shading).

of uncertainty is illustrated in figure 2 which gives the feasible regions for IMPC (light shading) and RPMPC (dark shading). Unsurprisingly, IMPC has a larger feasible region since no restrictions are built in to accommodate for the model uncertainty (which of course implies this would be invalid for the real uncertain system). Nevertheless, the difference can be observed to be relatively modest and furthermore we note that, in this case, both the nominal and robust invariant sets of K_2 are identical.

B. Control Performance

Robustness: In order to demonstrate the robustness with respect to model uncertainty of RPMPC compared with IMPC, we choose an initial state $[-9\ 0.5]^{\mathrm{T}}$ lying within the feasible regions of both RPMPC and IMPC. System dynamics are chosen as the LTI system described by $[A(k)\ B(k)] = [A_2\ B_2], \forall k \geq 0$. It is clear from figure 3 that RPMPC (solid line) steers the system to the origin. On the other hand, IMPC (dotted line) steers the system not only outside the feasible region of RPMPC, but also

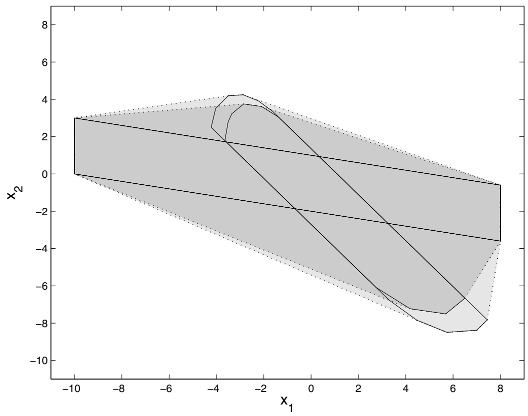

Fig. 2. Feasible region comparison of IMPC (light shading) and RPMPC (dark shading).

outside the nominal feasible region. This clearly indicates the necessity for deploying robust invariant sets and the efficacy with which RPMPC copes with model uncertainty.

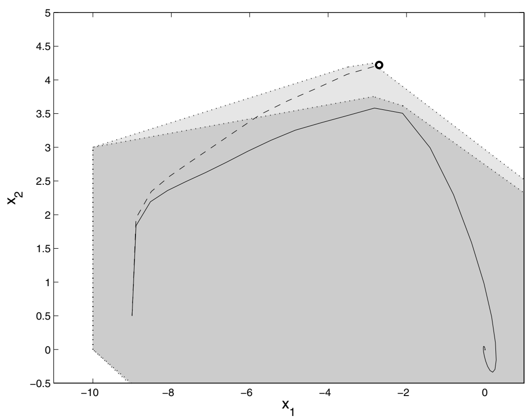

Fig. 3. Robustness comparison of IMPC (dashed) and RPMPC (solid).

Constraint handling: In a previous section it was already shown that RPMPC has a significantly larger feasible region than REMPC due to the fact that the real MAS of the controllers is used instead of an ellipsoidal inner approximation. A direct consequence is the improved ability to handle non-symmetrical constraints. To demonstrate this we take two initial states $[7.5\ -2]^{\mathrm{T}}$ and $[-7.5\ 2]^{\mathrm{T}}$ positioned symmetrically with respect to the origin and compare the behaviour of REMPC and RPMPC. System dynamics are chosen to be time-varying but identical for the different simulations. Figure 4 shows state and input trajectories of both methods for both initial states. As expected REMPC (dashed lines) produces identical state and input trajectories for both initial states (neglecting the sign) and hence handles the input constraints conservatively. RPMPC though (solid lines), can handle non-symetrical constraints efficiently and hence gives different shaped trajectories for each initial point; for instance, RPMPC gives input values above 0.5 for the initial state $[7.5\ -2]^{\mathrm{T}}$ but not for initial state $[-7.5\ 2]^{\mathrm{T}}$. Moreover, RPMPC is able to choose input values closer to the imposed input constraints. REMPC achieves a control cost of 586.9 for both initial states, while RPMPC results in costs 395.7 and 332.0 for intitial states $[-7.5\ 2]^{\mathrm{T}}$ and $[7.5\ \ -2]^{\mathrm{T}}$ respectively. This indicates that the improved constraint handling can lead to a significantly lower control cost.

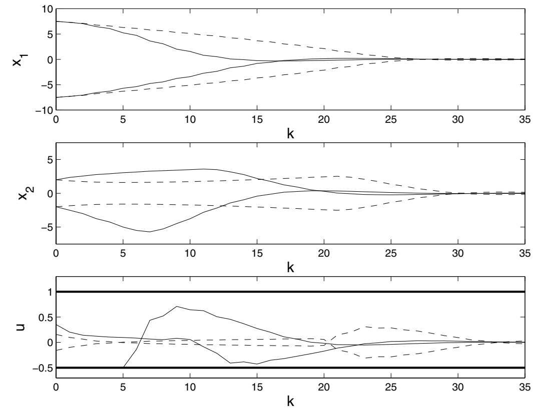

Fig. 4. Comparison of constraint handling of REMPC (dashed) and RPMPC (solid).

Optimality: In the previous subsection it was shown that the improved constraint handling can lead to significant improvements in control cost. In order to get a better overview of the overall difference in control behaviour and optimality between RPMPC and REMPC, we take 16 initial state values on a grid within the feasible region of REMPC and compare the performance and trajectories resulting from both algorithms. From figure 5 one can see that the trajectories are markedly different and with an average total control cost per simulation of 111.4 versus 164.4 one can conclude that RPMPC performs significantly better than REMPC in this example. The same can be expected to be the case for other examples.

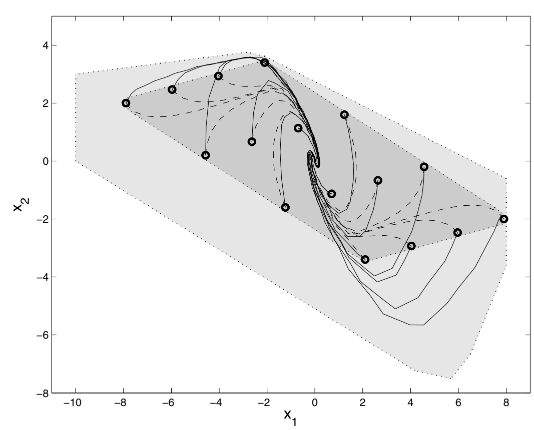

Fig. 5. Comparison of control behaviour of REMPC (dashed) and RPMPC (solid) for 16 different initial states within the feasible region of REMPC.

C. Computational Load

One might expect the computational load to increases significantly with the move from the nominal to the uncertain case. However, in fact RPMPC is expected to have a similar computational complexity to IMPC: both are formulated as a QP, have the same number of d.o.f. in the on-line optimization and have constraints determined by the number of inequalities describing the invariant sets (the plot of these sets in figure 2 suggests that the number of facets does not differ significantly). This observation is in direct contrast to a comparison with REMPC, an alternative robust algorithm. REMPC requires an SDP which is in general much more expensive than QP. In this example the average computational load per iteration for the first 50 iterations of all simulations depicted in figure 5 is 0.0062 sec. for RPMPC compared to 0.0404 sec. for REMPC, which is a difference of almost an order of magnitude. No specific attempts were made to optimize the computational efficiency of either implementation; simulations were done on a P4-2GHz PC running Matlab 6.5 using the standard toolboxes.

VI. Conclusion

In this paper a new interpolation based MPC algorithm using polyhedral robustly invariant sets is introduced for LPV systems with polytopic model uncertainty. The new method makes use of a new algorithm proposed in [11] that allows the efficient construction of polyhedral robustly invariant sets for LPV systems. The method guarantees recursive feasibility and robust asymptotic stability for a set of initial states that is equal to the convex hull of the invariant sets of a set of off-line selected stabilizing controllers. The method is illustrated on a numerical example and shown to have improved feasibility and control performance than robust interpolation based MPC using ellipsoidal invariant sets. Additionally, it is shown to be able to effectively stabilize LPV systems, unlike the nominal interpolation based algorithm using polyhedral invariant sets. Finally, it is worth pointing out that the new algorithm can efficiently cope with non-symmetrical state and input constraints, which other robust interpolation based MPC schemes typically cannot.

VII. Future Work

Interesting future research directions are the extension of the new algorithm to LPV systems subject to bounded disturbances. This would potentially be possible by making use of minimal admissable sets for LPV systems subject to disturbances, which is still an open research direction.

Another interesting research direction is the application of multi parametric quadratic programming to construct explicit piecewise affine solutions to the QP formulation of the new algorithm, which could potentially lead to an improved computational efficiency.

Finally, an interesting path is finding interpolation methods that do not need [10] the explicit decomposition of the states implied in (13), 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.

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. assistant with the I.W.T. (Flemish Institute for Scientific and Technological Research in Industry) at the Katholieke Universiteit Leuven. Universiteit Leuven, Belgium. the Katholieke Universiteit Leuven, Belgium.

References

[1] M. Bacic, M. Cannon, Y. I. Lee, and B. Kouvaritakis. General interpolation in MPC and its advantages. *IEEE Transactions on Automatic Control*, 48(6):1092–1096, June 2003.

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

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

[4] 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.

[5] E. G. Gilbert 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*, 36(9):1008–1020, 1991.

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

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

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

[9] 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.

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

[11] B. Pluymers, J.A. Rossiter, J. A. K. Suykens, and B. De Moor. The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty. *Accepted for publication in proceedings of the Americal Control Conference (ACC05), Portland, USA*, 2005.

[12] J.A. Rossiter, B. Kouvaritakis, and M. Bacic. Interpolation based computationally efficient predictive control. *International Journal of Control*, 77(3):290–301, 2004.

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

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

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

[16] 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.