Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 04-144

A Multi Parametric Quadratic Programming Solution To Robust Predictive Control [1]

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

March 2005

Proceedings of the IFAC World Congress 2005, Prague, Czech Replublic.

[1]This report is available by anonymous ftp from *ftp.esat.kuleuven.ac.be* in the directory *pub/sista/pluymers/reports/ifac05b_rmpc_mpqp.ps*

[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

Multi parametric quadratic programming is an alternative means of implementing conventional predictive control algorithms whereby one transfers much of the computational load to off-line calculations. This paper demonstrates how one can formulate a robust MPC problem as a quadratic program and hence make it amenable to MPQP solutions. The paper then derives some MPQP solutions and discusses the efficacy of these.

A MULTI PARAMETRIC QUADRATIC PROGRAMMING SOLUTION TO ROBUST PREDICTIVE CONTROL

J.A. Rossiter [**] **B. Pluymers** [*] [1] **J.A.K. Suykens** [*]
B. De Moor [*] [2]

[*] *Research Group SCD-SISTA, Department Of Electrotechnical Engineering (ESAT), Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium*
{bert.pluymers,johan.suykens,bart.demoor}@esat.kuleuven.ac.be
[**] *Department of Automatic Control and Systems Engineering, Mappin Street, University of Sheffield, S1 3JD, UK*
j.a.rossiter@sheffield.ac.uk

Abstract: Multi parametric quadratic programming is an alternative means of implementing conventional predictive control algorithms whereby one transfers much of the computational load to offline calculations. This paper demonstrates how one can formulate a robust MPC problem as a quadratic program and hence make it amenable to MPQP solutions. The paper then derives some MPQP solutions and discusses the efficacy of these. *Copyright© 2005 IFAC*

Keywords: Parametric programming, quadratic programming, robust MPC

1. INTRODUCTION

In recent years Multi parametric quadratic programming (MPQP) (Bemporad *et al.*, 2002) has become a useful tool for characterising, offline, the model predictive control (MPC) algorithm. The effectiveness of this approach is due to several factors: (i) the online implementation reduces to a look up table and hence is easy to code and (ii) the structure of the control strategy is transparent, which is not the case when using an online optimisation. Naturally there are limitations and much of the current research is investigating ways of simplifying the computation and complexity of the solution as these are significant barriers to applications with many states.

This paper however is concerned with a different aspect, that is robustness. Much of the works in the literature (Bemporad *et al.*, 2001; Borrelli *et al.*, 2001; Rossiter, 2004; Tondel *et al.*, 2003; Johansen *et al.*, 2003) concentrate on the linear case (hybrid and continuous time systems are not the topic of this paper). Yet if one looks at the MPC literature outside of the topic of MPQP, much of the current emphasis is on either robustness issues e.g. (Kothare *et al.*, 1996; Kouvaritakis *et al.*, 2000; Kerrigan, 2000) or non-linear systems. What makes this dichotomy seem even more surprising is that the very power of MPQP is its ability to handle the heavy computations offline and it is the need for burdensome optimisations that has severely limited the progress of robust MPC towards implementable algorithms; in fact this is recognised in some of the current offerings (Kouvaritakis *et al.*, 2000; Kothare *et al.*, 2003) which sacrifice some optimality and feasibility in order to deliver a simpler implementation.

Hence this paper has the purpose of considering how one might use the potential of MPQP to implement a robust MPC law. The solution may seem to be relatively obvious, but the application to robust MPC has specific advantages in the

[1] B. Pluymers is a research assistant with the I.W.T. Flanders.

[2] Research partially supported by Research Council KULeuven : GOA-Mefisto 666, FWO projects : G.0240.99, G.0197.02, G.0491.03, Belgian Federal Government : IUAP V-22 (2002-2006).

sense that the number of MPQP regions is shown to rise less steeply than the number of constraints in the QP formulation and hence facilitates online implementation.

The main reason why MPQP is difficult to apply to robust systems is that one needs to capture a whole class of predictions as opposed to just the nominal. Although this may seem tractable for a few steps, the combinatorial explosion (Schuurmanns *et al.*, 2000) of the possible predictions makes this intractable for even relatively low horizons and of course the usual thinking within MPC (Mayne *et al.*, 2000; Rossiter, 2003) is that large horizons are to be prefered in general. Of course there are ways around this, for instance works concentrating on disturbance rejection (Kerrigan, 2000) make use of one-step sets and backwards prediction as such an approach avoids the combinatorial difficulties.

However, many implementations nevertheless tend to rely upon the definition of an invariant and feasible terminal region, that is a space in which the trajectories for all possible uncertainties will remain once they have entered and of course, satisfy constraints. The existence of such a target region (Scokaert *et al.*, 1998; Rossiter *et al.*, 1998) allows one to restrict the active prediction horizon, n_c, to the same length as the number of free control moves; usually a low number and hence manageable even with a combinatorial growth in possible trajectories. Unfortunately, in the case of uncertain systems, most commonly modelled as linear parameter varying (LPV) systems, authors have generally assumed that maximal volume invariant terminal regions are not computable easily and have pursued other avenues. Typical routes in the literature assume that the invariant regions are ellipsoidal (which implies that the MPC algorithm becomes a semi-definite program (Cannon *et al.*, 2001)) or low complexity polyhedrals (Cannon *et al.*, 2002) (which severely restricts the feasible region); moreover these works often assume symmetric constraints. However, the good news is that a recent work (Pluymers *et al.*, 2004) demonstrated that, under certain conditions, it is possible to compute a maximal volume polyhedral terminal region for an LPV system; this result underpins the developments presented here.

This paper therefore unites several concepts:

(1) The use of a terminal region for robust MPC to keep the active prediction horizon small and hence to handle the combinatorial growth of possible trajectories.
(2) The use of backwards prediction in order to define the terminal region.
(3) Once the control law has been established as deploying a QP optimiser, it can immediately be solved as an MPQP.
(4) The method is not restricted to symmetric constraints (a typical assumption in many papers) and hence extends feasibility extensively.

The paper is organised as follows. Section 2 gives some background to the formulation of the robust MPC algorithm (Pluymers *et al.*, 2004), section 3 dicusses the MPQP implementation for the robust and nominal algorithms and section 4 gives some numerical illustrations of the differences. Section 5 contains conclusions.

2. BACKGROUND

2.1 A nominal MPC algorithm

This paper will assume discrete LPV state space models

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

with the nominal model being (A, B). The 'predicted' control law (Rossiter *et al.*, 1998; Scokaert *et al.*, 1998) takes the form

$$\begin{array}{ll} u_k = -Kx_k + c_k & k = 0, ..., n_c - 1 \\ u_k = -Kx_k & k \geq n_c \end{array} \quad (2)$$

where only the first move of this is ever implemented; c_k are d.o.f. available for constraint handling and thus are chosen as $c_i = 0$ when this is feasible. Performance, either predicted or actual, will be assessed by the cost

$$J = \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k \quad (3)$$

Substituting a nominal model and 'optimal' control law (2) into (3) and ignoring terms that do not depend upon the d.o.f., one can show (Rossiter, 2003) that the cost to be minimised, J_c, takes the form

$$J_c = C^T W_D C; \quad C = [c_0^T, ..., c_{n_c-1}^T]^T \quad (4)$$

where W_D =diag$(W, ..., W)$, and $W = B^T \Sigma B + R$, $\Sigma - \Phi^T \Sigma \Phi = Q + K^T R K$.

Assume that the process is subject to constraints:

$$\underline{u} \leq u_k \leq \overline{u}; \quad \underline{x} \leq x_k \leq \overline{x}, \qquad k = 0, \ldots, \infty \quad (5)$$

Then the maximal admissible set [1] (MAS) S_0 (Gilbert *et al.*, 1991) can be defined as:

$$S_0 = \{x : M_0 x \leq d_0\} \quad (6)$$

The maximal control admissible set (MCAS) S_c, the set from which in at most n_c moves one can drive the state into S_0, again without violating constraints, takes the form:

$$S_c = \{x : \exists C \text{ s.t. } M_0 x + N_0 C \leq d_0\} \quad (7)$$

[1] The largest invariant set in which the state trajectories do not violate constraints.

Finally therefore, a popular MPC algorithm with guarantees of recursive feasibility and convergence, for the nominal case, is given as:
MPC Algorithm: At each sampling instant, minimise the performance index:

$$\min_{C} \quad J = C^T W_D C \quad \text{s.t.} \quad M_0 x + N_0 C \leq d_0 \quad (8)$$

where only the first block element of C is implemented.

2.2 RMPC: MPC for the LPV case

Before moving onto the MPQP implementation, it is next necessary to derive a robust MPC (RMPC) algorithm deploying just an QP optimisation. For ease of presentation, we summarise the result first and the remainder of this section gives more detail which the reader could choose to skip.

2.2.1. The key result An RMPC algorithm can be summarised by the optimisation:
RMPC algorithm: At each sampling instant, minimise the performance index:

$$\min_{C} \ J = C^T W_D C \text{ s.t. } M_r x_k + N_r C \leq d_r \quad (9)$$

where only the first block element of C is implemented.

It has been argued (Pluymers *et al.*, 2004) that one can define inequalities such that the control law implied in (9) has:

(1) Recursive feasibility for system (1).
(2) Guaranteed stability and convergence for all initial points inside S_r (which is the MCAS for RMPC),

$$S_r = \{x : \exists C \text{ s.t. } M_r x_k + N_r C \leq d_r\}. \quad (10)$$

For completeness define the associated MAS as S_v, that is

$$S_v = \{x : M_r x \leq d_r\} \quad (11)$$

Remark 2.1. RMPC of (9) differs from MPC of (8) only in the inequalities that need to be satisfied, or in other words the replacement of S_0 by S_v and S_c by S_r. The key to the success of RMPC therefore is in the definition of these terms, that is, $M_r,\ N_r,\ d_r$.

2.2.2. Outline of algorithm development Next we give a brief overview of why algorithm (9) is robustly stabilising in the presence of constraints (5); for more details see (Pluymers *et al.*, 2004).

Performance index: This is based on nominal performance but in particular is phrased in terms of perturbations to the nominal control law. If one knows that the nominal feedback is stabilising for the LPV system (1), then one can prove convergence by establishing that the perturbation term C is decreasing. Given recursive feasibility, this can be determined from standard Lyapunov arguments.

Recursive feasibility: The key to (9) is the requirement that

$$x_k \in S_r \quad \Rightarrow \quad x_{k+1} \in S_r \quad (12)$$

and in fact to be more precise, one requires that an augmented state including the tail of C remains feasible, that is:

$$[M_r, N_r] \begin{bmatrix} x_k \\ C_k \end{bmatrix} \leq d_r \ \Rightarrow \ [M_r, N_r] \begin{bmatrix} x_{k+1} \\ C_{tail|k} \end{bmatrix} \leq d_r \quad (13)$$

where $C_{tail|k} = [c^T_{k+1|k}, ..., c_{k+n_c-1|k}, 0]^T$. Such a guarantee can be established by defining $M_r,\ N_r,\ d_r$ as the combination of several constraints:

(1) constraint matrices that ensure constraint satisfaction for all possible state trajectories over n_c steps.
(2) a constraint that ensures all possible n_c step ahead predictions enter the MAS S_v.

The precise details of how to compute these matrices are omitted to save space and are available in (Pluymers *et al.*, 2004).

3. THE MPQP IMPLEMENTATION

3.1 The structure of an MPQP algorithm

MPQP (Bemporad *et al.*, 2002) gives a parameterisation of the solution to a QP in terms of the variables which affect the cost and constraints but are not degrees of freedom. In the case of (8,9) this parameter is the state x. It is known that the optimum solution (in this case for C) has a piecewise affine (PWA) dependence on the state. So for instance, specific solutions in a given region take the form:

$$x \in S_i \ \Rightarrow \ C = -K_i x + p_i; \ \ S_i = \{x : M_i x \leq d_i\} \quad (14)$$

and the overal solution is defined over the union of all the regions S_i (which do not overlap). Moreover it can be shown that this union is equivalent to the MCAS S_r, that is:

$$S_r = \bigcup_i S_i \quad (15)$$

Remark 3.1. One of the regions will be the MAS (S_v); for this region $K_i = p_i = 0$.

Remark 3.2. The definitions of $M_i,\ d_i,\ K_i,\ p_i$ depend in a non-simple way upon the parameters of (8,9) the original optimisation (for instance $W_D,\ M_r,\ N_r,\ d_r$). As such one may find a relatively small change in the inequalities gives substantial changes to the shape and number of regions S_i as well as the corresponding $K_i,\ p_i$.

3.2 Differences between the nominal and robust algorithms

A quick glance is sufficient to note that optimisations (8) and (9) are of the same structure but differ in the inequalities. It is therefore expected that the solutions may look quite different. This in itself is very insightful because a simple plot can be used to demonstrate how the control law changes, with the state position, to accomodate the model uncertainty. Such plots are given in the next section.

Another interesting issue is complexity. The 'nominal' inequalities used in (8,9) are very different in dimension (no. or rows) due to the combinatorial nature of the predictions for the robust case. Although some simplification maybe possible offline by removing redundant rows, it would still be unsurprising if the robust algorithm did not have many more rows and hence could potentially [2] require significantly longer solution times for the online implementation. The hope, however, might be that the MPQP solution may not be substantially more complex. In this case one would be able to handle the robust problem with little more effort than the nominal case and that would be a significant benefit.

4. NUMERICAL ILLUSTRATIONS

The purpose of this section is simply to compute the MPQP (Kvasnica *et al.*, 2003) solution for the robust MPC algorithm, as this in itself is novel and then to compare it to the nominal MPC algorithm. It is perhaps self evident that RMPC will be more robust, although this is also illustrated, so our prime interest is to contrast the complexity of the two solutions so that some comment can be made about the potential of MPQP solutions to RMPC.

For simplicity of illustration and to aid comparison with other publications, we use the two state, double integrator example used by many other authors. The model takes the form of (1) where

$$A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{16}$$

$$A_2 = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \tag{17}$$

and $A(k) = (1-\lambda(k))A_1 + \lambda(k)A_2$, $B(k) = (1-\lambda(k))B_1 + \lambda(k)B_2$, $0 \le \lambda(k) \le 1$. For the purposes of this section to ensure consistency of comparisons, $\lambda(k)$ is time-varying but defined to be the same sequence for all simulations. The nominal model is chosen as $A = 0.5(A_1+A_2), B = 0.5(B_1+B_2)$. Constraints are taken as

$$\overline{u} = 1; \quad \underline{u} = -0.5; \quad \overline{x} = -\underline{x} = [10,\ 10]^T. \tag{18}$$

[2] Although one could argue that in general the number of rows is less significant a factor than the number of d.o.f..

The unconstrained feedback controller $K = [0.4558\ \ 0.3698]^{\mathrm{T}}$ is the LQR-optimal for $Q = \mathrm{diag}(1, 0.01), R = 3$.

4.1 Number of constraints vs number of d.o.f.

We first illustrate how the complexity of the underlying optimisations (8) and (9) differ by presenting, in table 1, the number of inequalities required to define S_c, S_r for various n_c. Unsurprisingly, RMPC requires significantly more inequalities, because it is capturing all possible predictions, not just the nominal. This could cause a marked slow down in online implementation times of the associated QP optimisation.

	Number of inequalities	
n_c	MPC	RMPC
1	13	17
2	15	31
3	19	56
4	21	102
5	24	194

Table 1. Comparison of inequalities

4.2 MPQP complexity comparisons

Next we consider, in table 2, the complexity of the MPQP solutions, that is how many regions are required to cover the MCAS. It is clear that the complexity of the MPC solution is less than for RMPC and moreover there seems to be a trend that the the number of regions for RMPC increases more rapidly with n_c. Nevertheless, when compared to the number of inequalities (table 1) the MPQP solution of RMPC relative to MPC could still be argued to be more favourable than the QP comparison; for instance at $n_c = 5$ the relative number of regions is approximately 3 whereas the relative number of inequalities is approximately 7.5.

	Number of regions	
n_c	MPC	RMPC
1	14	16
2	42	68
3	73	124
4	102	204
5	131	313

Table 2. Numbers of regions

4.3 Feasible region comparisons

For completeness, although not directly aligned to MPQP, it is important to compare the feasible regions (or MCAS). Hence Figures 1a,b,c overlay the MCAS for MPC (solid line) with RMPC (shaded) for $n_c = 0, 2, 5$ respectively. Unsurprisingly, the robust algorithm has more restricted

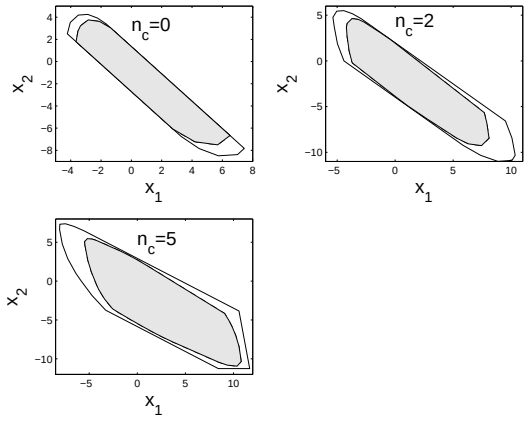


Fig. 1. Feasible regions with $n_c = 0, 2, 5$.

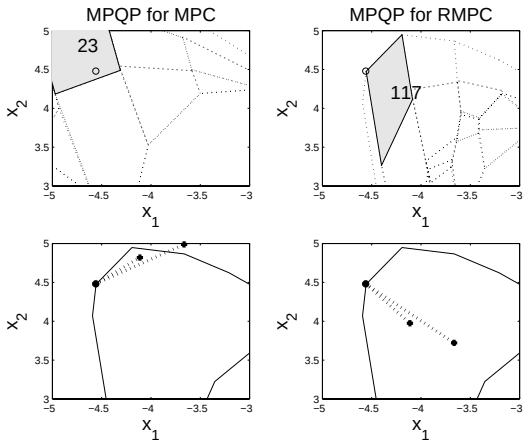


Fig. 2. MPQP regions and closed-loop predictions.

feasibility to account for all possible state evolutions, or in other words the nominal algorithm is in danger of suggesting solutions which lead to infeasibility (are outside S_r) and hence could fail.

4.3.1. Behaviour comparisons Next, we explore in more detail how the control strategy changes for a given region. To this end, we display in figure 2a,b and for $n_c = 3$, the MPQP regions for a specific part of the state space. The state $x = [-4.56, 4.48]^T$ (marked with a 'o') is in region 117 for RMPC and region 23 for MPC (regions are shaded). The corresponding control laws use:

$$\begin{array}{l|l} RMPC & c_0 = [1.35 \ \ -0.319]x + 6.66 \\ MPC & c_0 = [0.146 \ \ -0.114]x + 1.09 \end{array} \qquad (19)$$

which are clearly very different. We then overlay on the figure 2c,d the cone of one step ahead trajectories (marked by a dashed line and '+') that would arise for one-step ahead prediction. It is clear that these are very different for the two control laws with the MPC predictions (figure 2c) exiting the feasible region S_r whereas the RMPC predictions (figure 2d) remain inside and are obviously directed towards S_v.

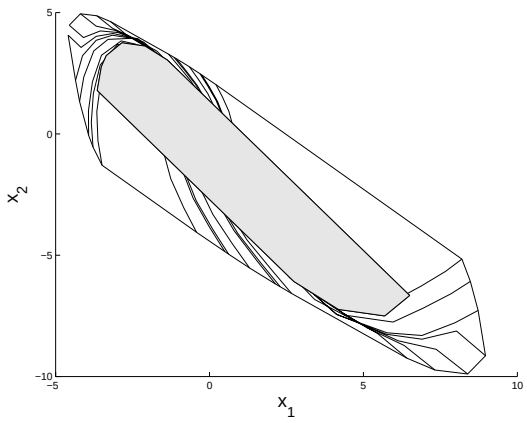


Fig. 3. RMPC closed-loop trajectories.

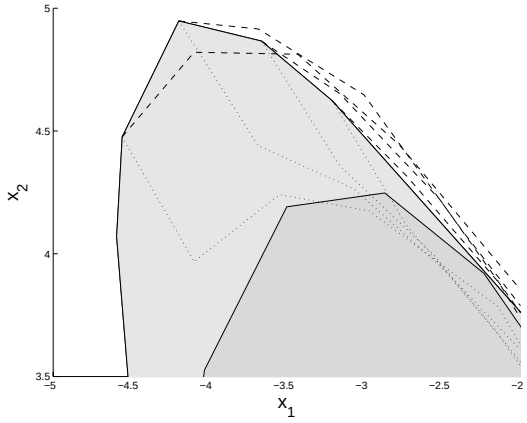


Fig. 4. RMPC and MPC closed-loop trajectories.

4.4 Closed-loop simulations

Finally, for completeness, we illustrate the behaviour of the MPQP solution to RMPC over the whole state space. Figure 3 presents the closed-loop trajectories for a number of initial conditions on the boundary of S_r ($\lambda(k)$ is the same for all these simulations but unknown and time-varying and the shaded area is S_v). What is clear is that all these trajectories remain within the feasible region S_r.

The corresponding simulations for MPC do not all remain within the feasible region. This is demonstrated in figure 4 (again with S_r, S_v shaded) where MPC is the dashed lines and RMPC is dotted; clearly the MPC trajectories move in a very different direction and notably some of these exit the feasible region.

5. CONCLUSIONS

This paper has demonstrated that MPQP solutions of reasonable complexity can be applied to robust MPC and in particular give a guarantee of convergence and recursive feasibility for a LPV system. We believe this is the first paper giving such results.

Some points of particular interest are:
(1) MPQP gives a explicit view of how the solution changes to take account of the model uncertainty. Such a view may be very insightful (for instance see figures 2,4).
(2) The single example here seems to show a more rapid increase in the complexity of the MPQP solution with the number of d.o.f. than for the nominal case (table 2). We intend to investigate whether this appears to be generic or specific to the given example.
(3) Robust MPC is often considered an intractable or very complex problem due to the combination of constraint handling with the combinatorial explosion in the number of possible predictions. However, it has been shown here that, in some cases, the MPQP solution is only a little more complex than for an equivalent linear problem and hence is an avenue worth further investigation.

It is also worth reiterating that the results given here are for non-symmetric constraints. Hence the increase in the volume of the feasible region beyond that which ellipsoidal based methods can achieve is very significant.

REFERENCES

Bemporad. A. and Filippi, C. (2001). Suboptimal explicit MPC via approximate multiparametric quadratic programming. *Proc. CDC*.

Bemporad, A., M. Morari, V. Dua and E.N. Pistokopoulos, 2002, The explicit linear quadractic regulator for constrained systems, *Automatica*, 38(1), 3-20.

Blanchini, F., 1999, Set invariance in control, *Automatica*, **35**, 1747-1767.

Borrelli, F., Baotic, M., Bemporad, A. and Morari, M., 2001, Efficient on-line computation of constrained optimal control. *Proc. IEEE CDC*.

Cannon, M., B. Kouvaritakis and J.A. Rossiter, 2001, Efficient active set optimisation in triple mode MPC, *IEEE Trans. AC*, 46(8), 1307-1313.

Cannon, M., Deshmukh, V., Lee, Y.I., and B. Kouvaritakis, 2002, Enlargement of polytopic terminal region in NMPC by interpolation and partial invariance, Proc. IEEE CDC.

Gilbert, E.G. and K. T. Tan, 1991, Linear systems with state and control constraints: the theory and application of maximal output admissable sets, *IEEE Trans. AC*, 36, 9, 1008-1020.

Grieder, P., Borrelli, F., Torrisi, F.D. and Morari, M., 2004. Computation of the constrained infinite time linear quadratic regulator. *Automatica*, 40(4):701–708.

Imsland, L., Bar, N. and B. Foss, 2004, A new algorithm for efficient MPC and a comparison with competing schemes, Proc. ACC.

Johansen, T.A. and Grancharova, A., 2003, Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Trans. AC*, 48(5):810–815.

Kerrigan, E., 2000, Robust Constraint Satisfaction: Invariant Sets and Predictive Control, Ph.D Thesis, Cambridge

Kothare, M.V., Balakrishnan, V. and Morari, M., 1996, Robust constrained model predictive control using linear matrix inequalities, *Automatica*, 32, 1361-1379.

Kouvaritakis, B., J.A. Rossiter and J. Schuurmans, 2000, Efficient robust predictive control, *IEEE Trans. AC*, 45(8), 1545-1549.

Mayne, D.Q., Rawlings, J.B., Rao, C.V. and Scokaert, P.O.M., 2000, Constrained model predictive control: stability and optimality, *Automatica*, 36, 789-814.

Kvasnica, M., P. Grieder, M. Baotic and M. Morari, 2003, Multi Parametric Toolbox (MPT), http://control.ee.ethz.ch/mpt,

Pluymers, B., Rossiter, J.A., Suykens, J.A.K. and De Moor, B., 2005, The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty, ACC

Pluymers, B., Rossiter, J.A., 2005, A simple algorithm for robust MPC, IFAC world congress

Rossiter, J.A., M.J. Rice, and B. Kouvaritakis, A numerically robust state-space approach to stable predictive control strategies. *Automatica*, 38(1):65–73, 1998.

Rossiter, J.A., 2003, Model based predictive control, CRC press

Rossiter, J.A. and Grieder, P., 2005, Using interpolation to improve efficiency of multiparametric predictive control. *Automatica*, 41,4.

Schuurmanns, J., and J.A. Rossiter, 2000, Robust piecewise linear control for polytopic systems with input constraints, *IEE Proc. CTA*, 147, 1, pp13-18

Scokaert, P.O.M. and Rawlings, J.B., 1998, Constrained linear quadratic regulation, *IEEE Trans. AC*, 43, 8, pp1163-1168.

Tøndel, P., Johansen, T.A. and Bemporad, A., 2003, Evaluation of piecewise affine control via binary search tree. *Automatica*, 39(5):945–950.

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