EFFICIENT MODEL PREDICTIVE CONTROL WITH PREDICTION DYNAMICS

Stian Drageset, Lars Imsland and Bjarne A. Foss[*]

Dept. of Eng. Cybernetics, Norwegian Univ. of Science and Technology, 7491 Trondheim, Norway.
fax: (+47) 73 59 43 99
e-mail: {drageset,lars.imsland,bjarne.a.foss}@itk.ntnu.no

Keywords: model predictive control, constrained systems, invariant sets, bilinear matrix inequalities

Abstract

An efficient robust model predictive control (MPC) strategy using augmented ellipsoidal invariant sets is enhanced by introducing more general dynamics in the future control moves (referred to as *prediction dynamics*) than simple time recession. This allows significant enlargements of the region of attraction for a give control horizon. As control horizons down to a single time step can be used, this implies online efficiency gains without decreasing performance. The procedure renders the offline problem a bilinear matrix inequality (BMI), thus algorithms for solving the non-convex offline problem is discussed.

1 Introduction

Model predictive control (MPC) has gained significant popularity in industry as a tool to optimise system performance while handling constraints explicitly. However, limitations on robustness and computational efficiency have restricted the application range. Early linear MPC algorithms [14, 10] guarantee nominal closed-loop stability by enforcing the state at the end of a given control horizon to be within a polytopic, feasible and invariant set. This requires the solution of a quadratic program (QP). An MPC algorithm that is robustly stabilising to certain classes of model uncertainty by applying semi-definite programming online is presented by [6]. However, both strategies quickly lead to computational intractability for large control horizons, which is frequently required to ensure initial feasibility.

An efficient MPC strategy stable to similar model uncertainties as treated in [6], but with online computational requirements linear to the control horizon is presented in [8]. The online efficiency is achieved by considering feasibility offline. Flexibility of the control input over the control horizon is still permitted by using an autonomous state-space formulation where the state vector is augmented by the degrees of freedom in future control input. Stability is ensured by enforcing the membership of the augmented state vector to an invariant and feasible ellipsoid at current time. For convenience this algorithm will be referred to as efficient predictive robust control, ERPC. The major drawback of ERPC is that the region of permissible initial values is conservative with respect to the original input/state constraints.

In this paper the autonomous state-space formulation of the offline problem of [8] is generalised. More degrees of freedom are added by incorporating prediction dynamics in the autonomous system. The price to pay is that this turns the convex linear matrix inequalities (LMI) problem of [8] into a non-convex BMI problem. The prediction dynamics allow future control to vary indefinitely with any control horizon, so in this respect the new approach gives the advantage of an infinite control horizon (although in a suboptimal manner) at the computational cost of short control horizons. In addition to enlargening the maximal range of initial values it turns out the new approach can increase online efficiency as shorter control horizons (down to just one time step) gives ellipsoids and performance comparable to those reported in [7].

The model class and control objective are introduced in Section 2, while Section 3 treats previous work on MPC using ellipsoidal invariant sets. Next, a specialised case of prediction dynamics is introduced and simulation results presented (Section 4). The general case of prediction dynamics, its interpretation, stability and a solver for the offline problem are presented in Section 5, while the results are discussed in Section 6.

2 Model class and control objective

Consider the discrete state-space model subject to input and state constraints

$$x_{k+1} = A_k x_k + B_k u_k \tag{1a}$$

$$\text{subject to } -\overline{u} < u_k < \overline{u}, \;\; \underline{x} < x_k < \overline{x}, \tag{1b}$$

with dimensions $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$. It will be assumed that perfect state knowledge is available and that the system is stabilisable. For uncertain systems a polytopic uncertainty model could be applied,

$$[A_k, B_k] \in \mathrm{Co}\left\{[A^1, B^1], \cdots, [A^n, B^n]\right\}, \tag{2}$$

where Co denotes the convex hull, and A^j and B^j are corners of the uncertainty set. However, for brevity a linear time invariant representation will be used, i.e. $A_k = A$, $B_k = B$. The generalisation to the polytopic uncertainty model is straightforward [8, 1].

The control objective will be to minimise (while satisfying constraints) the infinite horizon linear quadratic (LQ) cost function,

$$J_{LQ} = \sum_{i=0}^{\infty} x_{k+i+1}^{\mathsf{T}} Q x_{k+i+1} + u_{k+i}^{\mathsf{T}} R u_{k+i}, \tag{3}$$

where Q and R are positive semi-definite, symmetric matrices and x_{k+i+1} and u_{k+i} denote predicted[1] values of states and control inputs. We want the controller to be stabilising for a

[*] Corresponding author.

[1] Throughout the paper, we will not always use notation that distinguish predicted variables from the "real" variables.

set of initial conditions (region of attraction) that is as large as possible, but in doing this, we must sacrifice optimality for some initial conditions.

In the following the system will be prestabilised by a feedback controller [13], K, optimal with respect to (3) in the unconstrained case. Consider expressing the degrees of freedom as the perturbation, c_k, away from this optimal control, and let the future predicted control input in the MPC controller be

$$u_i = \begin{cases} Kx_i + c_i, & i = k, \dots, k + n_c - 1 \\ Kx_i, & i \geq k + n_c \end{cases} \tag{4}$$

As a consequence the optimisation is carried out in terms of the new free variables c_i. The c_i's are minimised, but must be large enough to prevent constraint violation. After the control horizon n_c, we can set $c_i = 0$ assuming the optimal LQ control is feasible. The system equation for (1) with (4) is

$$x_{k+1} = \Phi x_k + Bc_k, \tag{5}$$

where $\Phi = A + BK$.

3 Background

This section reviews some of the results of [8, 7].

3.1 Augmented state-space formulation

Consider the autonomous state-space model created by augmenting the state vector with the future degrees of freedom. Letting f denote the vector of future perturbations away from optimal control, c_k, over the control horizon, n_c, the dynamics of (5) can be described as

$$z_{k+1} = \Psi z_k \tag{6a}$$

where $z \in \mathbb{R}^{n_x + n_u n_c}$ and

$$z_k = \begin{bmatrix} x_k \\ f_k \end{bmatrix}, \; f_k^\mathsf{T} = \begin{bmatrix} c_k^\mathsf{T} & c_{k+1}^\mathsf{T} & \cdots & c_{k+n_c-1}^\mathsf{T} \end{bmatrix} \tag{6b}$$

$$\Psi = \begin{bmatrix} \Phi & [B\ 0\ \cdots\ 0] \\ 0 & M \end{bmatrix}, \; M = \begin{bmatrix} 0_{n_u} & I_{n_u} & 0_{n_u} & \cdots & 0_{n_u} \\ 0_{n_u} & 0_{n_u} & I_{n_u} & \cdots & 0_{n_u} \\ & & \ddots & & \\ 0_{n_u} & \cdots & 0_{n_u} & 0_{n_u} & I_{n_u} \\ 0_{n_u} & \cdots & 0_{n_u} & 0_{n_u} & 0_{n_u} \end{bmatrix}. \tag{6c}$$

Here z_k is the augmented state vector, Ψ is the augmented transition matrix, I_{n_u} is the identity matrix, and 0_{n_u} is a matrix of zeros, both of dimension $n_u \times n_u$. The M matrix simply accommodates the time recession of f_k. Note that the structure of M ensures

$$c_{k+n_c+i} = 0 \; \forall \; i \geq 0. \tag{7}$$

3.2 Offline problem: maximal invariant set

Feasibility is handled offline by finding an estimate of the region of attraction, the maximal set of initial values that gives stability. Stability is ensured by requiring the current augmented state, and hence implicitly all the predicted augmented states, to be within an invariant feasible ellipsoid $E_z := \{z | z^\mathsf{T} Q_z^{-1} z \leq 1\}$. The projection of this ellipsoid onto the state space (see [8]), $E_{xz} := \{x | x^\mathsf{T} (TQ_zT^\mathsf{T})^{-1} x \leq 1\}$ where T is defined by $x = Tz$, should be as large as possible. The volume of the ellipsoid is proportional to $\log\det(TQ_zT^\mathsf{T})$ [1]. This gives rise to the following constrained minimisation problem

$$\min_{Q_z} \log\det(TQ_zT^\mathsf{T})^{-1} \tag{8a}$$

$$\text{subject to} \quad \begin{bmatrix} Q_z & Q_z\Psi^\mathsf{T} \\ \Psi Q_z & Q_z \end{bmatrix} \geq 0 \tag{8b}$$

$$\bar{u}_j^2 - [K_j^\mathsf{T}\ e_j^\mathsf{T}] Q_z [K_j^\mathsf{T}\ e_j^\mathsf{T}]^\mathsf{T} \geq 0, \; j = 1, \dots, n_u \tag{8c}$$

$$F(Q_z) \geq 0 \tag{8d}$$

where Q_z is a symmetric positive definite matrix, e_j is the jth column of the identity matrix, and $\bar{u}_j$ and K_j correspond to the jth input. The invariance requirement (8b) is expressed as an LMI, (8c) ensures feasibility with respect to input constraints, and (8d) (if present) should be interpreted as an affine function representing ellipsoidal or polytopic "state constraints" (that is, constraints that put restrictions on the shape of E_{xz}). The optimisation problem is convex, and effective algorithms exist [1].

3.3 Online problem: minimising cost

As future control flexibility is part of the current augmented state, the ellipsoidal stability constraint can be applied at current time rather than at the end of the control horizon. This reduces online optimisation to minimising a performance index based on the future degrees of freedom in the input, J_f, subject to membership of the precomputed ellipsoid

$$\min_f J_f \quad \text{subject to} \quad z^\mathsf{T} Q_z^{-1} z \leq 1. \tag{9}$$

Here, J_f penalises the future control perturbations,

$$J_f = \sum_{i=0}^{n_c-1} c_i^\mathsf{T} W c_i \tag{10}$$

where W is given by

$$W = B^\mathsf{T} P B + R$$
$$P = Q + K^\mathsf{T} R K + \Phi^\mathsf{T} P \Phi.$$

It can be shown that J_f and the LQ cost (3) differ by a bias term, thus minimising the two indexes is equivalent [7]. The ellipsoidal constraint in (9) leads to suboptimality. This can be reduced to negligible levels by allowing a line search outside the ellipsoid, which will be referred to as scaling [7].

To summarise, the online minimisation is reduced to norm minimisation subject to a single quadratic constraint, which can be solved extremely efficiently.

3.4 Algorithm

We can now present an overall algorithm for ERPC.

Algorithm 1 (ERPC)
Offline: *Choose an optimal unconstrained K. Find the optimal invariant set, E_{xz} by solving (8). If the required set of initial values $X_0 \subset E_{xz}$, proceed to Step 2. If $X_0 \nsubseteq E_{xz}$, increase n_c.*

Online: *Perform the minimisation (9) and implement $u_k = Kx_k + c_k$, where c_k is the first element of f, and move on to the next time step.*

The feasible, invariant set and the stabilising K render the algorithm robustly asymptotically stabilising [8].

3.5 Conservative region of attraction

Whereas ERPC enjoys superior efficiency compared to traditional QP MPC approaches in terms of computation time, the constraint region is conservative, that is the ellipsoidal region of initial conditions is smaller than the maximal admissible region for the augmented system (6), which is a polytope created by linear constraints [5]. This might severely limit the admissible initial conditions, and hence exclude practical applications.

Two recent approaches expand the region of attraction, but both reduce efficiency. In [9] suboptimality is avoided by using an invariant polytopic set. However, the increased complexity of the region requires a linear programming (LP) solver whose computational demands are comparable to those of convex quadratic programming (QP). Triple mode MPC of [12] uses a regular quadratic programming MPC for the first few time steps.

4 Case study: introducing prediction dynamics

The main idea in this paper is to let the matrix M be a general matrix instead of having the “time recession” structure of the previous section. We will refer to this as “prediction dynamics”.

This section introduces prediction dynamics for single input systems with a control horizon of one time step. This turns the “non-convex parameter” of the offline problem into a scalar, rendering the offline problem readily solvable by line search. Restating (6) we get

$$\begin{bmatrix} x_{k+1} \\ c_{k+1} \end{bmatrix} = \begin{bmatrix} \Phi & b \\ 0 & m \end{bmatrix} \begin{bmatrix} x_k \\ c_k \end{bmatrix},$$

where the M matrix (6c) that shifts the future control freedom c_k, is reduced to a scalar, m. When m is allowed to vary, the optimisation problem (8) becomes a bilinear matrix inequality (BMI) problem. However, stability considerations confine m to the interval $[-1, 1]$. This allows m to be used as a tuning parameter instead of n_c, where m is fixed at each iteration and a line-search can be used to find the m that gives sufficiently large area. We will refer to this as the Line-search Offline Problem, LOP. For the single input case, minimising J_f is equivalent to minimising $\|f\|^2$ as m is a scalar. Using m as a tuning parameter instead of n_c has the significant advantage that the online computation time is constant for different tuning parameters.

Example 1 Consider the following state-space model with input constraints [7]:

$$A = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix} \tag{11a}$$

$$-1 \le u \le 1, \tag{11b}$$

for which the optimal LQ gain with $Q = \mathrm{diag}(1, 0)$ and $R = 0.1$ is $K = -[2.828,\ 2.826]$.

First, we investigate how varying m in the offline problem increases the region of attraction. Figure 1 shows how the ellipsoidal volume varies with m (solid line). The area is measured by $\log\det(TQ_zT^{\mathsf{T}}))$ as in the optimisation problem of [8]. The dashed line that illustrates the area for $m = 0$ (which equals ERPC for $n_c = 0$), indicates that using an m on the interval $0 < m < 1$ produces larger areas than ERPC. The dotted line illustrates the area for ERPC with $n_c = 10$. For $m > 0.83$ LOP with $n_c = 1$ gives a larger area than ERPC does with $n_c = 10$. Figure 2 illustrates that the ellipsoids of LOP (solid) with $m = 0.85$, 0.95 and 0.97 are much larger than the ellipsoids of ERPC (dotted) with $n_c = 10$, 20 and 30. In fact the superiority of LOP is greater than the figure reveals, increasing m further easily gives ten times larger ellipsoids than Figure 2 [3], whereas our implementation of ERPC breaks down due to numerical difficulties for $n_c > 30$. However, as Figure 2 indicates, the larger ellipsoids tend to get narrower and do not necessarily contain the sets produced by ERPC as subsets. As the ellipsoids are tilted as m varies, the union of ellipsoids for different m values covers large parts of the second and fourth quadrants for the given example.

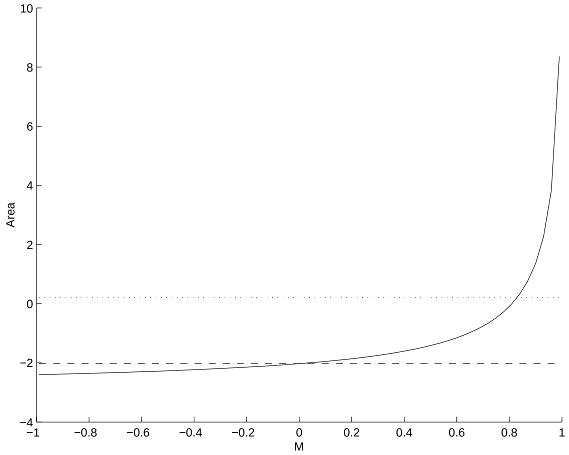

Figure 1: Area of invariant sets for Example 1: LOP (–), ERPC for $n_c = 0$ (- -) and $n_c = 10$ ($\cdots$).

To summarise, the region of attraction compared with ERPC is increased. In fact, the prediction dynamics (which can be said to allow control freedom beyond the control horizon) allows LOP to produce larger regions of attraction than the maximal admissible set [5] for the augmented system (6) for the same control horizon. In addition to the improved region of attraction, LOP improves efficiency and to some degree performance, at least for this example. The increased online computational efficiency is due to ERPC requiring a larger n_c to include a given set of initial conditions. Experiments showed that the online implementation of LOP is about three times faster than ERPC for $n_c = 10$. Of course, this efficiency superiority increases if larger regions of attraction and hence larger n_c is required for ERPC.

Table 1 shows a minor performance improvement for a random pair of initial conditions using the line search offline problem approach, measured by a suitably adapted cost index (10). However, when the algorithms are extended with scaling [7],

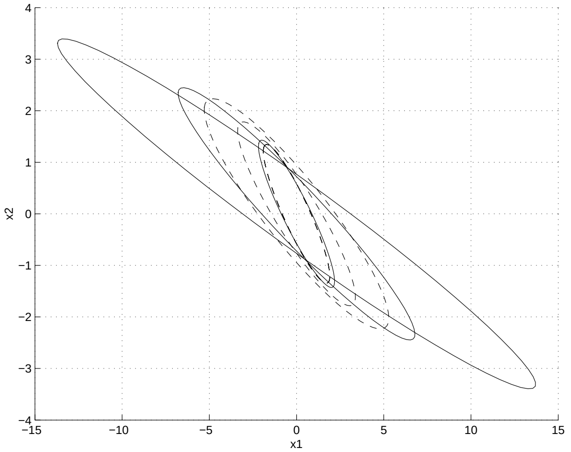

Figure 2: Invariant ellipsoids for ERPC (- -) and LOP (–). ERPC is plotted for $n_c = 10$, 20 and 30 and LOP is plotted for $M = 0.85$, 0.95, and 0.97.

the difference is negligible.

Table 1: Cost for Example 1: ERPC with $n_c = 10$ and LOP with $n_c = 1$ and $M = 0.855$

	Without scaling		With scaling	
Initial position	ERPC	LOP	ERPC	LOP
$x_a = -[0.4\ \ 0.3]^{\mathsf{T}}$	3.5088	3.4561	3.3499	3.3499
$x_b = [-1.4\ \ 0.5]^{\mathsf{T}}$	15.7375	15.5839	14.8773	14.8773

Example 2 The successful application of the LOP approach to a nonlinear system described by a polytopic model was demonstrated by experiments on a nonlinear model of an inverted pendulum [3]. The nonlinear system was represented by a linear difference inclusion, where state constraints were used to restrict the system state to the domain on which the inclusion is valid. The computations and simulations support the results from Example 1.

5 General case: prediction dynamics and BMIs

5.1 Offline problem as BMIs

This section investigates the general case (several inputs and control horizon longer than one step) of prediction dynamics, BMIMPC. Consider system (6), with M in (6c) a general matrix. The offline problem (8) remains unchanged except that Ψ is now a variable as it contains M, which renders the offline problem a (non-convex) BMI problem. As M is no longer a scalar as in the previous section, a multivariable approach will be devised. We will refer to this offline problem as BMIOP. BMIOP is a generalisation of LOP as it solves the same problem as LOP when $n_c = 1$. The online problem remains the same as for ERPC, and is solved by the online part of Algorithm 1.

5.2 Effect of prediction dynamics

Let us investigate the effects of the new degrees of freedom. The update equation is

$$f_{k+1} = M f_k \tag{12}$$

$$c_k = \begin{bmatrix} I_{n_u} & 0_{n_u} & \cdots & 0_{n_u} \end{bmatrix} f_k =: D f_k \tag{13}$$

where M initially is defined as in (6c). If an element of M is perturbed, the definition of f_k as in (6b) no longer applies. This makes the interpretation that each element of f specifies a future open-loop input invalid for BMIMPC.

For ERPC, $f_{k+n_c} = M^{n_c} f_k = 0$, as the eigenvalues of M are zero. This means it is implicitly required that the optimal LQ feedback is feasible after the control horizon, as

$$c_{k+i} = 0 \ \forall \ i \geq n_c. \tag{14}$$

However, inserting non-zero diagonal or sub-diagonal elements into M gives non-zero eigenvalues and non-zero f_{k+n_c}. This implies that c_i, and hence the feedback, can vary indefinitely, even with a control horizon of only one time step. The input is kept more restrictive than the LQ feedback as long as required by the constraints, independently of n_c. In effect, one could say the tuning of n_c is done implicitly, and instead M is tuned to get sufficiently large invariant sets.

Interestingly, this means that BMIMPC tunes the region of attraction without affecting online computational requirements. The result is that we in a way have the control freedom of a large control horizon at the computational cost of a much shorter one. But it should be noted that this effect of a longer control horizon is sub-optimal, since it must obey the dynamics of M. However, experiments have shown that tuning with M can give significantly larger ellipsoids than tuning with n_c only.

If only super-diagonal elements of M are perturbed, the eigenvalues of M remain unchanged, (14) still applies and as expected, examples show negligible changes on the size of the region of attraction.

Using a general M means that in general c_i is non-zero after the end of the control horizon n_c. This requires the performance index (open loop cost) J_f to be altered, by letting $n_c \to \infty$. Standard arguments show that this infinite sum can be expressed as a quadratic function,

$$J_f = \sum_{i=0}^{\infty} c_{k+i}^{\mathsf{T}} W c_{k+i} = f_k^{\mathsf{T}} \Gamma f_k \tag{15}$$

where Γ is positive definite given by $M^{\mathsf{T}} \Gamma M - \Gamma = D^{\mathsf{T}} W D$.

5.3 Algorithm and stability: BMIMPC

The general BMIMPC approach can be summarised as follows:

Algorithm 2 (BMIMPC)
Offline *Choose an optimal unconstrained K. Find n_c, M and Q_z that give a suitable invariant set, e.g. by using LOP or BMIOP.*
Online *Perform online part of Algorithm 1*

The stability proof of Algorithm 2 is similar to that of Algorithm 4.1 of [8]:

Theorem 3 (Closed loop stability) *If for system (1) there exist K, Q_z, n_c and M such that $x_k \in E_{xz}$, the closed-loop application of BMIMPC is feasible and robustly asymptotically stabilising.*

Proof. Invariance of E_{xz} ensures $x_{k+i} \in E_{xz} \forall\ i > k$, which by an induction argument guarantees feasibility for all future time steps. Stability follows as the stabilising control law $u = Kx$ is used in a region around the origin [8].

To prove convergence, note that $\tilde{f}_{k+1} = Mf_k$ is feasible and $\|\tilde{f}_{k+1}\|_\Gamma^2 = \|f_k\|_\Gamma^2 - c_k^\mathsf{T} W c_k$. Minimising $\|f_{k+1}\|_\Gamma$ gives $\|f_{k+1}\|_\Gamma^2 \leq \|f_k\|_\Gamma^2 - c_k^\mathsf{T} W c_k$, from which we conclude that the sequence $\|f_k\|_\Gamma$ converges (it is monotonically decreasing and lower bounded). Since $W > 0$, this means c_k must converge to zero, which means that Algorithm 2 reverts to the control law $u = Kx$, which is convergent. ■

5.4 Solving the BMIs

An augmented Lagrangian approach to sequential semi-definite programming [4], which optimises all variables simultaneously, was chosen for solving the BMIOP. The approach, extracts the non-convexity of (8b) by including an equality constraint in the merit function, Fletcher's augmented Lagrangian [11],

$$L(P_z, Q_z, \Lambda) = \log(\det(TQ_zT^\mathsf{T})^{-1}) + \mathrm{tr}(\Lambda(P_zQ_z - I)) + \frac{\nu}{2}\,\mathrm{tr}(P_zQ_z - I)^\mathsf{T}(P_zQ_z - I), \quad (16)$$

where Λ is a Lagrange multiplier matrix, ν is a penalty term, tr denotes matrix trace, $\mathrm{tr}(P_zQ_z - I)$ expresses the constraint violation of the extracted equality, and I is the identity matrix of dimension $n_x + n_c$. At each iteration we find the search direction p^i for a line-search update of the current value of w^i (a vector with the elements of Q_z and P_z), by minimising a quadratic approximation of (16),

$$\nabla L^\mathsf{T} p^i + \frac{1}{2}(p^i)^\mathsf{T} \nabla^2 L p^i, \quad (17)$$

where ∇L denotes the gradient of L and $\nabla^2 L$ the (convexified) Hessian. Formulated as a semidefinite program, this gives the following minimisation,

$$\min_{p^i, t} t \quad \text{subject to} \quad \begin{bmatrix} t - \nabla L^\mathsf{T} p^i & (p^i)^\mathsf{T} \\ p^i & 2\nabla^2 L^{-1} \end{bmatrix} \geq 0 \quad (18a)$$

$$w^i + p^i \in X_{LMI} \quad (18b)$$

where the quadratic objective is transformed to an LMI constraint (18a) with a slack variable, t. The convex set given by X_{LMI} (18b) refers to the LMI constraints (8d) and (8c) in addition to the invariance LMI,

$$\begin{bmatrix} Q_z & \Psi \\ \Psi^\mathsf{T} & P_z \end{bmatrix} \geq 0, \quad (19)$$

which is equivalent to (8b) provided $P_z = Q_z^{-1}$.

The optimisation strategy is to temporarily allow the iterates to move away from the equality constraints to achieve rapid movement towards the optimum. Eventually the iterates are forced to approach the equality by the increasing penalty term, ν. The termination criteria make sure the solution fulfills the equality to a sufficient accuracy. The tuning of the Lagrangian and penalty term is important to obtain good performance, see [4] for suggestions.

A significant advantage is that exact expressions for the gradient and Hessian of the merit function are available. The gradient and Hessian of the $\log\det(\cdot)^{-1}$ function are

$$\frac{\partial}{\partial A}\log\det A^{-1} = -A^{-1}, \ \frac{\partial^2}{\partial A^2}\log\det A^{-1} = A^{-1} \otimes A^{-1}$$

where $\otimes$ is the Kronecker product. The gradient and Hessian of the two last parts of the merit function are the same as in [4].

Merely using optimality constraints as termination criteria is not necessarily successful when not having added state constraints. This is because overly large ellipsoids rarely will be a desired solution, as they tend to become narrow and have poor numerical conditioning. Experience shows that checking feasibility in addition to convergence of $\|P_zQ_z - I\|$ is important.

Lastly, it must be clear that this formulation of the problem is non-convex, and any solution found by the above procedure is local, and hence not guaranteed to be globally optimal. Also, second order algorithms such as the one above typically requires initial conditions not too far from a solution. Using the ERPC solution as initial condition remedies this, and also ensures that the solution we find is as least as good as ERPC (in terms of size of the ellipsoid).

6 Discussion

6.1 Comparison of LOP and ERPC

Simulations on two examples showed that the LOP approach enjoys two important advantages over ERPC: It produces significantly larger regions of attractions and improves efficiency. At the same time it achieves equal or better performance to ERPC. Example 1 was taken from a paper presenting ERPC [7], whereas Example 2 was independently chosen [3].

Let us consider a couple of potential objections to LOP's superiority on region of attraction. First, it can be argued that the main reason for the large difference in volume is caused by inadequacies of the semi-definite solver. However, an improved solver would of course improve LOP as well. More importantly, ERPC's numerical problems occur for control horizons that are so large that their online computational burden might preclude applications anyway.

Second, even though LOP's region of attraction is a lot larger, the shape is not necessarily the most convenient. Long, narrow ellipsoids might be impractical. However, a suitable orientation of an ellipsoid can be easier to find due to the LOP's increased flexibility (M varies continuously as opposed to n_c) and efficiency (n_c is small).

The inclusion of prediction dynamics complicates the offline problem only. LOP's online efficiency is superior as long as ERPC uses a control horizon longer than one time step. This follows as the online problem is dominated by computations with matrices that grow linearly or quadratically with n_c. The improved efficiency of LOP over ERPC is problem and implementation specific. Simulations indicate LOP is three times faster when $n_c = 10$.

Simulations without scaling gave information on cost differences (Table 1). It seems that for a given initial point the cost is larger when the ellipsoid boundary is far beyond this point as opposed to when it is close to the ellipsoid boundary. LOP was found to give better performance than ERPC for the ellipsoidal volumes that gave best performance. For the ellipsoids that resulted in worse performance for both algorithms ERPC produced the best results. On average the cost performance of the two algorithms seemed to be similar. For the initial values of [7] varying m improved the cost by between 1 and 2 per cent compared to using ERPC. For both the considered examples, extending the algorithms with scaling eradicated these differences.

6.2 Comparison of LOP/ERPC and BMIOP

BMIOP is applicable to both multiple input systems and larger control horizons, as it systematically deals with a multivariable M. To interpret the results of BMIOP, consider BMIOP as a hybrid of ERPC and LOP. ERPC tunes its region of attraction, ROA, by varying n_c, LOP by varying M, whereas BMIOP uses a combination of the two. For $n_c > 1$ the initial volume for BMIOP (calculated with M as specified for ERPC) will be larger than the initial volume for LOP. This means a smaller effect of varying M will be necessary for BMIOP to get the desired volume. Hence, for low n_c BMIOP resembles LOP closer, and for large n_c it resembles ERPC closer.

If operating regions are found where ERPC gives better cost performance than LOP, the cost characteristics of BMIOP can be made similar to ERPC by using the largest control horizon allowed by online time constraints, and if necessary the ROA can be increased further by optimising M.

The hybrid characteristics also allow BMIOP to combine the strengths of LOP and ERPC and give a superior ROA to both algorithms. Simulations indicate that the initial ROA, calculated by M as specified in ERPC, is contained as a subset when the ROA is enlarged by varying M. Recall that LOP did not include all ellipsoids of ERPC with high n_c as subsets, even though it produced larger ellipsoids. BMIOP with $n_c = 10$ will be expected to contain ERPC's ROA produced with $n_c = 10$ as a subset, but can be enlarged further just as LOP.

An easy way to achieve similar properties to BMIOP, is to allow perturbations of one single diagonal or sub-diagonal element of the M matrix. Examples have shown that this can give similar ROA and cost to BMIOP[3]. This approach can be interpreted as an algorithm for BMIOP with constraints on the structure of M, or a simple extension of LOP. Whether the potential gain of more freedom in M is large enough to justify the increased complexity of BMIOP is a topic for further research.

Triple mode MPC [12, 2] increases the region of attraction by using a traditional QP MPC strategy on the first few time steps, before switching to ERPC. The algorithm retains an efficiency advantage over standard QP dual mode MPC approaches as the control horizon used for QP is reduced. Of course, the inclusion of the computationally demanding QP renders the algorithm less efficient than ERPC and BMIMPC. If we interpret "triple mode MPC" as an outer shell that uses ERPC as a black box, the improvements of prediction dynamics apply equally to triple mode MPC as it does to ERPC.

7 Conclusion

An efficient MPC algorithm for systems with polytopic uncertainty is shown to be robustly asymptotically stabilising, similarly to ERPC of [8]. The inclusion of prediction dynamics has the potential to enlarge the region of attraction of ERPC, even with a control horizon of one time step. The efficiency gain is achieved at the cost of a more complex offline problem, without loss of online performance.

Acknowledgments

The authors would like to thank Dr. Mark Cannon for valuable comments on the implementation of ERPC. The second author acknowledges the NTNU Natural Gas Research Center for financial support.

References

[1] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Number 15 in SIAM Studies in Applied Mathematics. SIAM, 1994.

[2] M. Cannon, B. Kouvaritakis, Y. I. Lee, and A. C. Brooms. Efficient non-linear model based predictive control. *Internat. J. Control*, 74(4):361–372, 2001.

[3] S. Drageset. A bilinear matrix inequalities approach to efficient model predictive control. Master's thesis, Department of Engineering Cybernetics, NTNU, 2002. http://www.itk.ntnu.no-/ansatte/Imsland_Lars_Struen/BMIEMPC.pdf.

[4] B. Fares, P. Apkarian, and D. Noll. An augmented Lagrangian method for a class of LMI-constrained problems in robust control theory. *Internat. J. Control*, 74(4):348–360, 2001.

[5] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: the theory and application of maximal output admissible sets. *IEEE Trans. Aut. Control*, 36(9):1008–1020, 1991.

[6] M. V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–79, 1996.

[7] B. Kouvaritakis, M. Cannon, and J. A. Rossiter. Who needs QP for MPC anyway? *Automatica*, 38(5):879–884, 2002.

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

[9] Y. I. Lee and B. Kouvaritakis. Linear matrix inequalities and polyhedral invariant sets in constrained robust predictive control. *Internat. J. Robust Nonlinear Control*, 10(13):1079–1090, 2000.

[10] K. Muske and J. Rawlings. Model predictive control with linear models. *AIChE J.*, 39(2):262–287, 1993.

[11] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.

[12] J. A. Rossiter, B. Kouvaritakis, and M. Cannon. Computationally efficient algorithms for constraint handling with guaranteed stability and near optimality. *Internat. J. Control*, 74(17):1678–1689, 2001.

[13] J. A. Rossiter, B. Kouvaritakis, and M. J. Rice. A numerically robust state-space approach to stable-predictive control strategies. *Automatica*, 34(1):65–73, 1998.

[14] M. Sznaier and M. J. Damborg. Suboptimal control of linear systems with state and control inequality constraints. In *Proc. 26th IEEE Conf. Decision Contr.*, pages 761–762, Los Angeles, CA, 1987.