TIME VARYING TERMINAL CONTROL

L. Imsland [*,1] **J.A. Rossiter** [**]

[*] *Department of Engineering Cybernetics, Norwegian Univ. of Science and Technology, 7491 Trondheim, Norway*
[**] *Department of Automatic Control and Systems Engineering, Mappin Street, University of Sheffield, S1 3JD, UK*

Abstract: Two recent contributions to the literature (Kouvaritakis et al., 2000; Imsland et al., 2004) have shown how to use offline analysis to reduce online computation while enlarging the feasible regions of a control law. Both methods make use of an augmented system so this paper gives some proper discussion of their differences and similarities and in particular it gives new insight to the structure of the solutions. Following on the paper then discusses the potential of both methods and makes proposal for future developments.

Keywords: Model predictive control, linear systems, computational efficiency

1. INTRODUCTION

One of the conflicts within model predictive control (MPC) is how to obtain a large feasible region, that is the operating region within which the closed-loop input, output and state do not violate constraints, and at the same time retain optimum performance. The conundrum is that algorithms giving large feasibility regions often give relatively poor performance and vice versa. For instance it is well known that detuning the control law will result in smaller input variations and therefore inputs are less likely to violate constraints.

In practice, it is not clear what impact retuning might have on feasibility or performance and in fact one might argue that the formulation of an algorithm to do a systematic trade off is still an open question. A possible solution is to use a dual mode MPC algorithm (Scokaert and Rawlings, 1998; Rossiter, 2003), that is one which allows degrees of freedom for the first n_c control moves and then assumes some fixed (terminal) control law thereafter. Such an algorithm guarantees optimal performance near the origin, but feasibility is dependent on both the choice of the terminal law and n_c. In theory one could increase n_c as much as required to achieve the desired feasibility region, but large values of n_c are usually frowned on in the MPC community. The alternative of detuning the terminal law has an impact on performance.

Another issue of equal importance is how to ensure robust stability and feasibility. In this case the same conflicts arise but the more usual MPC algorithms using quadratic programming (QP) are not appropriate due to the difficulties of testing constraint satisfaction over the uncertain system [2]. As such authors turned to approaches based on ellipsoidal invariant sets as LMI solvers could be used efficiently for this case. An example is Efficient Robust Predictive Control (ERPC) (Kouvaritakis et al., 2000), for which an unexpected added bonus was that the online computation actually became much simpler and increasing n_c had only a relatively small impact, so higher values could be used.

The downsides of ERPC are the restriction to ellipsoidal regions, which is not a topic discussed here, and that the computational load and conditioning of the offline algorithm does not scale well with n_c. This is because the algorithm still

[1] Presently with SINTEF ICT, Applied Cybernetics, N-7465 Trondheim, Norway

[2] Recent work is beginning to tackle this limitation (Pluymers *et al.*, 2004)

adopts a conventional dual mode structure, n_c free moves followed by a fixed law. To overcome this limitation, other authors (Imsland et al., 2004) considered how to add more dynamics into the control solutions and hence to achieve gains in feasibility without recourse to large n_c; this they denoted GERPC. The purpose of this paper is to give better insight into the ERPC and GERPC algorithms and hence to propose sensible directions for further development.

Section 2 gives some background to the algorithms, Section 3 focuses on insight, Section 4 on illustrations and the paper ends with some proposals. We note that although robust control is a key motivation for the ERPC and GERPC algorithms discussed hereafter, it is not a key issue in the comparisons here.

2. BACKGROUND

This section gives some more details about dual mode MPC algorithms and in particular ERPC and GERPC. This is necessary background for the new insights and developments of Section 3.

2.1 Overview of linear MPC

Assume discrete state space models

$$x_{k+1} = Ax_k + Bu_k. \tag{1}$$

Define performance, either predicted or actual, by the cost

$$J = \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k. \tag{2}$$

Let the 'predicted' control law (Rossiter et al., 1998; Scokaert and Rawlings, 1998) be:

$$\begin{array}{ll} u_k = -Kx_k + c_k & k = 0, \ldots, n_c - 1 \\ u_k = -Kx_k & k \geq n_c \end{array} \tag{3}$$

where c_k are d.o.f. available for constraint handling. This formulation allows d.o.f. during transients and assumes a fixed state feedback in the asymptotic behaviour.

One can show that, for K the (unconstrained) optimal (Rossiter, 2003), J takes the form

$$J = C^T W_D C + p \tag{4}$$

where $C = [c_0^T, \ldots, c_{n_c-1}^T]^T$, $W_D = \mathrm{diag}(W, \ldots, W)$, $W = B^T \Sigma B + R$, $\Sigma - (A-BK)^T \Sigma (A-BK) = Q + K^T R K$. The term p is not dependent on the d.o.f. C and hence can be omitted.

Assume that the process is subject to constraints:

$$\underline{u} \leq u \leq \overline{u}; \quad \underline{x} \leq x \leq \overline{x}. \tag{5}$$

Then it can be shown the constraint satisfaction of the predictions for model (1) in conjunction with control law (3) is equivalent to membership of the the maximal controlled admissible set (MCAS), that is:

$$S_c = \{x : \exists C \text{ s.t. } M_0 x + N_0 C \leq d_0\}. \tag{6}$$

Definition of M_0, N_0, d_0 are omitted as standard but cumbersome. Also let the MAS be given as $S_0 = \{x : M_0 x \leq d_0\}$.

Finally, the MPC law is given by minimising J (4) subject to (6).

Algorithm 2.1. **MPC algorithm:** (Scokaert and Rawlings, 1998) At each sampling instant, perform the optimisation:

$$\min_C \quad J = C^T W_D C \;\; \text{s.t.} \;\; M_0 x + N_0 C \leq d_0. \tag{7}$$

Use the first block element of C in control law (3). Note that $x \in S_0 \;\Rightarrow\; C = 0$.

2.2 Conflicts for nominal MPC

The major conflict is between the volume of the feasible region S_c (6) and the achievable performance.

- If n_c is large enough (Scokaert and Rawlings, 1998), one can show that the MCAS is the largest feasible space possible and moreover the control law is the global optimum.
- In general, for computational (and sometimes robustness) reasons, n_c is chosen small.
- If n_c is small, then the volume of the MCAS maybe dominated by the implied state feedback K within (3), hence a highly tuned K could give rise to small MCAS and a lesser tuned K could give much larger feasible regions.
- Conversely, if K is poorly tuned, then the cost function is dominated by poorly performing predictions and hence the closed-loop control may also be severely suboptimal.

The designer has to get a balance between the volume of the feasible region S_c, the computational load (implied by n_c) and the implied performance (affected by K). There are currently no systematic tools for achieving this balance. Authors have therefore looked at ways of maximising the feasible region without sacrificing too much performance and while utilising a computational inexpensive optimisation. However, unsurprisingly, there is a hard limit on what can be achieved in this trade off when in essence, for a fixed n_c there is only one variable to play with, that is K. Moreover, changes in K change the shape as well as the volume of S_c and it can be hard to make precise judgements as to what is better.

2.3 ERPC

ERPC (Kouvaritakis et al., 2000) was formulated to deal with the robust case and hence was based on ellipsoidal invariant sets. Leaving aside this difference for now, this section outlines how the algorithm is set up to maximise the feasible region.

First augment the system (1) with m_c future control moves (the vector $f = [c_0^T, \ldots, c_{m_c-1}^T]^T$), to get an augmented model of the form:

$$z_{k+1} = \underbrace{\begin{bmatrix} A-BK & B \\ 0 & I_L \end{bmatrix}}_{\Psi} z_k; \; I_L = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ & & \ddots & & \\ 0 & \cdots & 0 & 0 & I \\ 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

with $z = \left[x^T, f^T\right]^T$. Then maximise the volume of the projection onto x-space of an invariant set for $z_{k+1} = \Psi z_k$, subject to z_k and the implied $u_k = -Kx_k + e_1^T f_k$ satisfying constraints (5). Let such an invariant set be denoted $\mathcal{E}_z$, given by

$$[x^T, f^T] \underbrace{\begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix}}_{Q_z^{-1}} \begin{bmatrix} x \\ f \end{bmatrix} \le 1. \qquad (9)$$

Now, the projection to x-space is given by

$$\mathcal{E}_x = \{\exists f : x^T P_{11} x \le 1 - f^T P_{22} f - 2x^T P_{12}^T f\},$$

or alternatively, f must be chosen such that:

$$(f - Hx)^T P_{22}(f - Hx) \le \\ 1 - x^T[P_{11} + H^T P_{22} H]x; \quad P_{22}H = -P_{21}.$$

Remark 2.1. The feasibility region is actually defined by a fixed linear state feedback (Rossiter et al., 2001). We notice that a feasible point for f can only exist when $x^T[P_{11} + H^T P_{22} H]x \le 1$, at which boundary one must have that $f = Hx$. That is, on the boundary of feasibility, there is a fixed dependence of f on x. The control law on the boundary is therefore given as $u = [-K + e_1^T H]x$. Let $K_{ERPC} = -K + e_1^T H$ and define the feasible invariant ellipsoid where $f = 0$ as:

$$\mathcal{E}_{x0} = \{x : x^T P_{11} x \le 1\}. \qquad (10)$$

Remark 2.2. The use of the vector $f = Hx_k$ as "initial condition" in autonomous model (8) defines a series of c_k which guarantee that x_{k+m_c} is within the invariant ellipsoid $\mathcal{E}_{x0}$. As with conventional dual mode MPC algorithms, the additional dynamics f are nil potent, that is they decay to zero after m_c steps.

2.4 GERPC

More recently some authors (Imsland et al., 2004) have suggested that the ERPC approach could be improved by augmenting the mode 2 assumption so that instead of implying nil potent dynamics to the state feedback, instead one could add extra dynamics which do not decay to zero in finite time. This change, that is the addition of extra dynamics, should give more d.o.f. for either increasing the volume of the implied terminal region or for improving predicted performance.

The algorithm is best illustrated as a change to the implied autonomous model (8) which governs the behaviour in mode 2,

$$z_{k+1} = \underbrace{\begin{bmatrix} A - BK & BD \\ F & G \end{bmatrix}}_{\Psi} z_k; \quad z = \begin{bmatrix} x \\ f \end{bmatrix} \qquad (11)$$

where one notes the addition of the terms D, F, G in lieu of the single term I_L. It was shown that one could formulate a BMI optimisation w.r.t. to the new variables D, F, G so that the projection of the invariant set for $z_{k+1} = \Psi z_k$ onto x-space was bigger than for ERPC[3]. Recent work in progress (Cannon et al., 2004) show that given $F = 0$ and $m_c \ge n_x$, an equivalent LMI (convex) optimisation can be formulated. Therein, it is also pointed out that the implied cost can be upper bounded by γ if the invariance condition $\Psi^T Q_z^{-1} \Psi - Q_z^{-1} \le 0$ is strengthened to

$$\Psi^T Q_z^{-1} \Psi - Q_z^{-1} \le -\frac{1}{\gamma} \begin{bmatrix} I & -K^T \\ 0 & D^T \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} I & 0 \\ -K & D \end{bmatrix} \qquad (12)$$

The γ is an effective tuning parameter for the performance vs. size of region conflict for GERPC.

Remark 2.3. The GERPC predictions are in fact single mode in that the pertubation term f converges to zero only asymptotically. Nevertheless, (given $F = 0$) the online optimisation would minimise a weighted norm of f at each sampling instant and moreover one could still choose $f = 0$ when $x \in \mathcal{E}_{x0}$ and hence recover optimal performance when close to the origin.

2.5 Triple mode MPC

Both ERPC and GERPC start from dual mode thinking, albeit GERPC arrives at a different solution. One alternative suggestion for overcoming the conflict between performance and feasibility is to allow more complex terminal control laws. So, instead of assuming a dual mode prediction structure such as in (3), some authors have looked instead at terminal controls such as:

$$\begin{array}{ll} u_k = -Kx_k + c_k & k = 0, \ldots, n_c - 1 \\ u_k = -K_{k-n_c} x_k, & k = n_c, \ldots, n_c + m_c - 1 \\ u_k = -Kx_k & k \ge n_c + m_c \end{array} \qquad (13)$$

where the notable change is the definition of terms $K_i,\ i = 0, \ldots, m_c - 1$ and hence the addition of a 3rd mode into the predicted control law.

The advantage of using structure (13) is manyfold:

(1) The predictions can still be constructed as having a linear dependence on the d.o.f. C
(2) The terminal region, with $C = 0$, may be significantly enlarged by the addition of the LTV part in (13). Hence the associated MCAS could also be much larger.
(3) The predictions still retain the 'optimal' feedback asymptotically and this helps ensure that the performance being minimised is still close to what we would ideally minimise given a higher n_c.

However, the weakness of Triple mode is the same as its strength, that is the structure implied in (13). It can be shown that the MAS (that is the feasible region for $C = 0$) depends strongly upon K_0 as the first implied control action within the predictions is always $u = -K_0 x$. Hence, the terminal region is still restricted to those that can be determined with a fixed state feedback. The advantage is just that one has essentially built into the predictions a gradual re-tuning of K as the

[3] Obvious as $D = 0, F = 0, G = I_L$ is a possible solution.

state moves nearer to the origin. Another major weakness is the current lack of a systematic tool for identifying the best sequence of K_i.

3. INSIGHTS AND EXTENSIONS

This section gives some new insights into the relationship between ERPC and GERPC. These insights are used to propose a GERPC based Triple mode algorithm with large feasibility regions which serves as a useful starting point for future development.

3.1 Feasible regions for ERPC and GERPC

It has been shown previously (Remark 2.1) that the feasible region of ERPC is equivalent to that which could be obtained for a fixed state feedback K_{ERPC}. We will show here that in fact a similar statement can be made for GERPC.

Theorem 3.1. The feasible region of GERPC is equivalent to that of a fixed state feedback.

Proof: The largest invariant ellipsoid for (11) can be represented as $\mathcal{E}_z = \{z : z^T Q_z^{-1} z \leq 1\}$. This is clearly the same form as (9) and hence the same analysis as Remark 2.1 must follow, giving a corresponding feedback K_{GERPC}. □

One might argue that Theorem 3.1 implies that choosing $m_c > n_x$ will give no increase in the feasible region. More rigorous proofs of this fact are in (Cannon et al., 2004).

3.2 Comparison of ERPC and GERPC

One might wonder that if both ERPC and GERPC have feasible regions restricted to that obtainable with a fixed state feedback, surely in some sense their feasible regions should be equivalent. Why would one algorithm be prefered to the other? This section attempts to give some insight into the different properties.

- ERPC is restricted to those strategies, which in at most m_c steps, move the state from its current position to a point where $x \in \mathcal{E}_{x0}$. This is evident from the structure of I_L.
- GERPC allows dynamics into the part of the model containing f, and f approaches zero only asymptotically. As a consequence the predicted control moves only approach $u = -Kx$ asymptotically and more importantly, one does not insist on the state moving to within $\mathcal{E}_{x0}$ in only m_c steps. It is this change which allows the feasible region to be enlarged.

Of course the weakness of GERPC is that $f \neq 0$ even when the predicted state may be well inside $\mathcal{E}_{x0}$ and this could give rise to some suboptimality. However, if $F = 0$, then $f = 0$ is feasible inside $\mathcal{E}_{x0}$ and you could obtain the optimal control.

A simplified comparison of GERPC and ERPC could reduce to the following:

(1) For large enough m_c, the feasible regions would be the same as both reduce to a fixed state feedback on the outer boundary (for which an invariance result on $\mathcal{E}_z$ is immediate and therefore convergence can be implied inside $\mathcal{E}_z$). However, for large m_c numerical problems might occur for ERPC (and GERPC, but for GERPC there is no advantage (in terms of size of ellipsoid) in choosing $m_c > n_x$ (Cannon et al., 2004)).
(2) For small m_c, GERPC allows more steps for x to move into $\mathcal{E}_{x0}$ and hence should be able to give larger $\mathcal{E}_x$.
(3) For the same m_c, GERPC requires in general a far more burdensome optimisation than ERPC. However, if $m_c \geq n_x$ and $F = 0$, then GERPC reduces to LMI optimisation with comparable complexity to ERPC.
(4) Performance optimisation can be built directly into ERPC by minimising $f^T W_d f$ subject to $z \in \mathcal{E}_z$. A similar comment apply to GERPC (Imsland et al., 2004). However, as ERPC assumes optimal feedback after n_c steps, the optimsation is better posed and hence better performance is likely. GERPC is optimising over trajectories which do not default back to the optimal and this mismatch will cause some suboptimality.

4. USING ERPC AND GERPC AS A BASE FOR TRIPLE MODE CONTROL

One major difficulty with the Triple mode algorithm of Section 2.5 is that there is no immediately obvious way of identifying the K_i used in Mode 2. This is because we are wanting the associated MAS S_i to be such that:

$$x_k \in S_i \text{ and } u_k = -K_i x_k \;\Rightarrow\; x_{k+1} \in S_{i+1}. \quad (14)$$

However, in practice the MAS are non-trivially defined and moreover K_i should ideally be selected to optimise performance. The search for a systematic algorithm is ongoing.

4.1 Using ERPC for Triple mode

In the interim, it has been noted (Rossiter et al., 2001) that in fact ERPC automatically produces a set of suitable K_i which satisfy (14) where S_i are defined as ellipsoidal invariant sets. That this is so can be taken immediately from Section 2.3 where it was shown that on the feasibility boundary, $f = Hx$. It is therefore implicit that one can write

$$f = \begin{bmatrix} c_0 \\ \vdots \\ c_{m_c-1} \end{bmatrix} = \begin{bmatrix} K_0 \\ \vdots \\ K_{m_c-1} \end{bmatrix} x = Hx.$$

The corresponding MAS (terminal constraint set for 1st mode) would be given directly from (6) as

$$S_{T0} = \{x : (M_0 + N_0 H)x \leq d_0\}$$

and the predicted cost for predictions (13) would be quadratic in $w^T = [C^T, x^T H^T]$.

4.2 Triple mode MPC with GERPC

GERPC (Imsland et al., 2004) can be used in a similar manner as above, but with further enhanced feasibility, and with a tuning parameter for the tradeoff between feasibility and performance. As the predictions for GERPC are not the same as for ERPC, some detail is warranted.

In order to formulate a triple mode algorithm along similar lines to that indicated in (13) we need information about the implied control law structure in the predictions. To obtain this we require the data Q_z (the matrix defining the augmented invariant ellipsoid $z^T Q_z^{-1} z \leq 1$, with size to some degree decided by choice of γ), the matrices D, F and G giving the augmented autonomous system (11) and matrix H corresponding to Q_z.

Second mode control moves. As for using ERPC, the feedback in the second mode is defined by $f = Hx_{n_c}$. This is used as initial condition to define future behaviour through the augmented system (11). The polytope where the above feedback is feasible (which x_{n_c} must be within), is found by projecting the MAS for the augmented system (11). For instance, on similar lines to equation (6) let the MAS for (11) with $f = Hx$ be given as $S_H = \{x : M_H x \leq d_H\}$. Since

$$x_{n_c} = (A-BK)^{n_c} x + \big[(A-BK)^{n_c-1}B, \ldots, B\big]C \quad (15)$$

the triple mode MCAS will be

$$S_{cH} = \{x : \exists C \text{ s.t. } M_H x + N_H C \leq d_H\}. \quad (16)$$

Control objective. The future cost is the cost of the first n_c control moves C, added to the cost of the infinite series of c_i's defined from the f in the autonomous model of (13). Consequently, the optimisation objective becomes $C^T W_d C + \sum_{i=n_c}^{\infty} (Df_{k+i})^T W D f_{k+i}$, where the future f_i's (along with predicted states) are given by (11). The infinite sum can be calculated by

$$\sum_{i=n_c}^{\infty} f_{k+i}^T D^T W D f_{k+i} = z_{k+n_c}^T \Gamma z_{k+n_c}, \quad (17)$$

where Γ is given by the Lyapunov equation $\Psi^T \Gamma \Psi - \Gamma = -[0\ D]^T W [0\ D]$.

The GERPC triple mode MPC algorithm can be summed up as follows:

Algorithm 4.1. For given x, minimize $J = C^T W_d C + x_{n_c}^T [I, H^T] \Gamma [I, H^T]^T x_{n_c}$ subject to $M_H x + N_H C \leq d_H$ and (15). Apply first block element of C as current input.

Stability follows similarly to triple mode MPC with ERPC (Rossiter et al., 2001).

Remark 4.1. The Mode 3 dynamic is not present in the predictions as f only tends to zero asymptotically. Thus, with this formulation, we are not guaranteed that the calculated control is the same as the unconstrained optimal even within $\mathcal{E}_{x0}$. This can be remedied by switching to the online formulation of GERPC within $\mathcal{E}_x$ (corresponding to what is done in Triple mode with ERPC in (Rossiter et al., 2001)), or switching to an "ordinary" dual mode MPC algorithm within the corresponding MCAS.

5. EXAMPLE

This section will illustrate the comparisons in feasibility and performance between GERPC and ERPC and Triple mode algorithms based on them for the system given by

$$A = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix} \quad (18)$$

subject to the constraints

$$|u_k| \leq 1, \quad |[0\ 1]\, x_k| \leq 1. \quad (19)$$

Choose $Q = \text{diag}(1, 0)$ and $R = 0.1$. When solving the Triple mode problem, we have chosen horizon $n_c = 5$ in the outer problem, and $m_c = 5$ degrees of freedom (horizon) in the "inner" ERPC problem. Since $m_c > 2$ does not achieve larger regions in GERPC, we have used $m_c = 2$ in GERPC and found the GERPC regions with the LMI algorithm from (Cannon et al., 2004) for different γ.

5.1 Elliposidal and polyhedral feasible regions

Fig. 1 shows the feasible regions for ERPC.

(1) Dark shaded region is the MAS for $u = -Kx$.
(2) Next lighter shading is MAS for Mode 2 of triple mode.
(3) Next lighter shading is MCAS for Triple mode.
(4) Lightest shading is MCAS for Algorithm 1 with $n_c = 10$.
(5) White line ellipsoid is for ERPC with $n_c = 5$.

Similar figures can be produced for GERPC, but with much larger regions. To save space, we instead include a comparison of MCAS for Triple mode for (a) ERPC, (b) GERPC with $\gamma = 1e2$ and (c) GERPC with $\gamma = 1e5$. See Fig. 2.

As expected the GERPC regions are much larger than for ERPC, and the γ parameter effectively tunes size of MCAS.

5.2 Performance

Simulations of state trajectories from initial condition (-1.75,0.5) can be seen in Fig. 1 (ERPC triple mode (solid), GERPC $\gamma = 1e2$ (dashed) and GERPC $\gamma = 1e5$ (dotted)). These trajectories are very different. Moreover, Fig. 3 shows that it takes much longer before the perturbation c becomes zero for GERPC.

The table below shows the runtime cost for both cases and again this demonstrates what was expected, that ERPC significantly outperforms GERPC for $\gamma = 1e5$ when it is feasible. Tuning

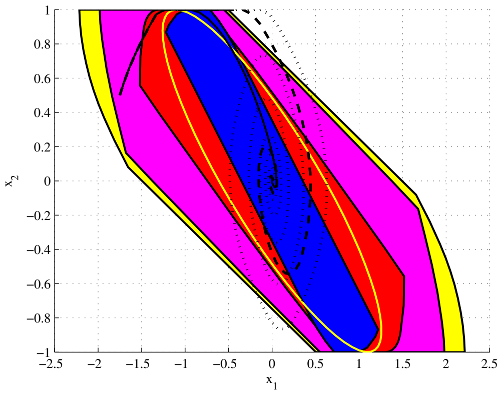

Fig. 1. Feasible regions for ERPC.

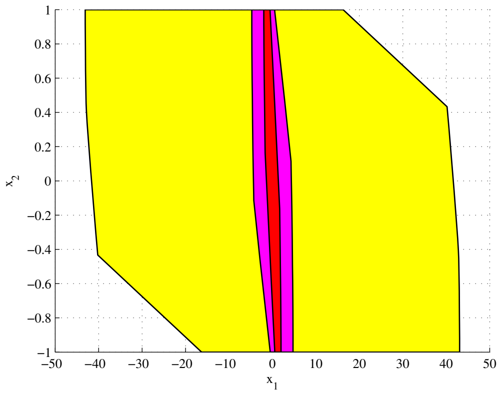

Fig. 2. MCAS for ERPC (smallest), GERPC w/$\gamma = 1e2$ and GERPC w/$\gamma = 1e5$ (largest).

Table 1. Runtime costs

ERPC	GERPC $\gamma = 1e2$	GERPC $\gamma = 1e5$
26.1	29.9	44.9

GERPC for better performance (reducing γ) gives smaller ellipsoids, but the cost gets closer to the ERPC cost.

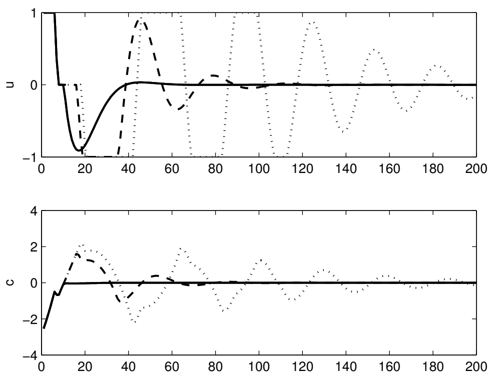

Fig. 3. Input u_k and c_k for Triple mode MPC with ERPC and GERPC ($\gamma = 1e2$ dashed, $\gamma = 1e5$ dotted), initial condition (-1.75,0.5).

6. FUTURE WORK AND CONCLUSIONS

This paper has given arguments and numerical illustrations to support the conjecture that GERPC will always give a larger feasiblity region than ERPC, but when feasible one would expect ERPC to give better performance. Another useful insight is that GERPC, just as ERPC, reduces to a fixed state feedback on the outer feasibility boundary; this observation has not been used to advantage in either algorithm and a systematic means of using this knowledge remains an open question. Also, based on this previous observation, it has been shown that, just as for ERPC, GERPC can be used as a base for a Triple mode algorithm and hence to give yet further feasibility improvements and the potential to recover some performance. The same comparative behaviour between GERPC and ERPC variants is expected.

A main purpose for discussing the insights of this paper is to consider a good direction for future developments. It is clear that the Triple mode algorithm based on ERPC or GERPC is still to some extent flawed for two reasons:

1) The mode 2 part of the prediction is based on a law chosen to maximise feasibility, not to optimise performance.

There needs to be some systematic compromise between the feasibility and performance and perhaps one such that is time (or state) varying so that one can change the underlying assumption as the state moves closer to the origin. A step in this direction is the use of an upper bound on GERPC performance (Cannon et al., 2004), as shown in the example.

2) The Triple mode algorithm is defined with polyhedral sets and as such does not lend itself easily to the robust case which ERPC and GERPC handle well.

We intend to look at how to bridge this gap and hope the recent work of (Pluymers *et al.*, 2004) will be a good start point.

REFERENCES

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

Cannon, M. and B Kouvaritakis, Optimising prediction dynamics for robust MPC, private communication.

Imsland, L., N. Bar, and B.A. Foss., 2004 A new algorithm for efficient MPC and a comparison with competing schemes. In *Proc. Amer. Contr. Conf.*, Boston, USA, 2004.

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

Pluymers, B., Rossiter, J.A., Suykens, J. and De Moor, B., 2004, The efficient computation of polyhedral invariant sets for linear systems with polytoic incertainty, submitted ACC

Rossiter, J.A., 2003, *Model-based Predictive Control - a Practical Approach.* CRC Press, 2003.

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

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

Scokaert, P.O.M., and J.B. Rawlings, 1998, Constrained linear quadratic regulation. *IEEE Trans. Aut. Control*, 43(8):1163–1169, 1998.