Robust Triple Mode MPC

Lars Imsland, J. Anthony Rossiter, Bert Pluymers and Johan Suykens

***Abstract*— This paper reviews triple mode predictive control for LTI systems, and proposes a new algorithm for robust triple mode predictive control for constrained linear systems described by polytopic uncertainty models. The approach significantly enlarges the feasibility region compared to robust dual mode approaches. The efficacy of the approach is demonstrated with numerical examples.**

I. Introduction

The use of predictive control for linear time invariant (LTI) systems subject to constraints is now well established [1, 2]. Standard guidelines to ensure guarantees of feasibility and/or stability are also commonly accepted, that is, many authors use the dual-mode prediction paradigm [3] in conjunction with an infinite horizon. Within this paradigm one assumes that the predictions revert to a linear closed-loop with a known and fixed linear feedback K after n_c steps. A major remaining obstacle is to balance (a) the desirable volume of the feasible region ($\mathcal{S}_c$, the set of states for which the controller can generate a feasible input also referred to as the Maximal Controlled Admissible Set, MCAS) with (b) complexity and hence the available computational power as well as (c) achievable performance for given n_c.

- If n_c is large enough [3], 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, 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 the closed-loop control may also be severely suboptimal.

The designer has to get a balance between the volume of the feasible region $\mathcal{S}_c$ (affected by K and n_c), the computational load (implied by n_c) and the implied performance (affected by K and n_c). There are currently no systematic tools for achieving this balance. Authors (e.g. [4, 5]) 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 $\mathcal{S}_c$ and it can be hard to make precise judgements as to what is better.

One suggestion that has been little considered in the literature is the concept of triple mode control [6]. In this strategy one recognises that large feasible regions in conjunction with good performance often imply nonlinear or linear time varying (LTV) prediction dynamics [7]. The challenge is to find a suitable and fixed LTV control law which enlarges feasibility without too much detriment to performance (it is now well known [8] that in general the optimal trajectory is piecewise affine (PWA) and hence even an LTV strategy will have limited benefits).

The first triple mode controller [6] used the algorithm of [9] to specify the additional mode of the MPC control law. In [9] ellipsoidal invariant sets were computed for a dual mode MPC setup. Recently, the generalisation of these results in [10, 11] was used in [12] to specify a more flexible triple mode algorithm, but still for the nominal case. However, as both [9] and [10, 11] originally are developed for the robust case, it is a natural next step to consider robust triple mode MPC algorithms. For this, one need algorithms for computing polytopic robustly invariant sets, which recently have been developed, and used for robust dual mode MPC [13, 14].

The paper is organised as follows. Section II gives some background to dual and triple mode paradigms for LTI systems, while Section III reviews some robust dual mode approaches for linear systems with polytopic uncertainty. These are used to define a new triple mode algorithm with robust stability guarantees in Section IV. The example in Section V confirms that the triple mode algorithm gives substantially larger feasibility region compared to its dual mode counterparts.

II. Background to Triple mode MPC

This section reviews a conventional MPC algorithm followed by a triple mode variant for the LTI case. Both are formulated using polyhedral invariant sets, and hence online optimization is based on QPs.

A. Dual mode 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

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

Let the 'predicted' control law [3, 15] be:

$$\begin{aligned} u_k &= -Kx_k + c_k, \qquad k = 0, \dots, n_c - 1, \\ u_k &= -Kx_k, \qquad\quad\ \ k \geq n_c, \end{aligned} \tag{3}$$

L. Imsland is with SINTEF ICT, Applied Cybernetics, N 7465 Trondheim, Norway. Lars.Imsland@sintef.no

J. A. Rossiter is with the Department of Automatic Control and Systems Engineering, Mappin Street, University of Sheffield, S1 3JD, UK. J.A.Rossiter@Sheffield.ac.uk

B. Pluymers is a research assistant and Johan Suykens is an associate professor with the IWT Flanders at the Katholieke Universiteit Leuven, ESAT-SCD-SISTA, Kastelpark Arenberg 10, B-3000 Leuven (Heverlee), Belgium. Bert.Pluymers@esat.kuleuven.be

where c_k are degrees of freedom available for constraint handling. This formulation allows d.o.f. during transients and assumes a fixed state feedback in the asymptotic behaviour. For K the (unconstrained) optimal [1], J takes the form

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

where $C = [c_0^{\mathsf{T}}, \ldots, c_{n_c-1}^{\mathsf{T}}]^{\mathsf{T}}$, $W_D = \mathrm{diag}(W, \ldots, W)$, $W = B^{\mathsf{T}}\Sigma B + R$, $\Sigma - (A-BK)^{\mathsf{T}}\Sigma(A-BK) = Q + K^{\mathsf{T}}RK$. 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:

$$L_x x + L_u u \leq l. \tag{5}$$

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

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

Definitions of M_0, N_0, d_0 are omitted as standard but cumbersome. Also let the maximal admissible set (MAS - feasible region with $u = -Kx$, [16]) be given as $\mathcal{S}_0 = \{x : M_0 x \leq d_0\}$.

Algorithm 2.1 (MPC algorithm, [3]): The MPC law is given by, at each sampling instant k, performing the optimisation:

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

Use the first block element of C in control law (3).
Note that $x \in \mathcal{S}_0 \;\Rightarrow\; C = 0$. We reiterate briefly that the conflicts in this control law are between the volume of the MCAS (given by (6)) and performance (given by (2)), which for a given n_c can give contrary requirements on K.

B. Triple mode MPC

One suggestion for overcoming the conflict between performance and feasibility is to allow more complex terminal control laws [6, 7]. For example, an LTV law may be close to the true PWA optimal law and also allows much larger feasibility and moreover implies a known 'exact' quadratic cost.

So, instead of the dual mode prediction structure of (3), some authors have proposed terminal controls such as:

$$\begin{array}{ll} u_k = -Kx_k + c_k, & k = 0, \ldots, n_c - 1, \\ u_k = -Kx_k + d_{k-n_c}, & k = n_c, \ldots, n_c + m_c - 1, \\ u_k = -Kx_k, & k \geq n_c + m_c, \end{array} \tag{8}$$

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

Assume that $d = [d_0^{\mathsf{T}}, d_1^{\mathsf{T}}, ..., d_{m_c-1}^{\mathsf{T}}]^{\mathsf{T}} = Hx_{n_c}$, that is the d_i values depend only on the x value at the commencement of mode 2. Then one can easily rework the mode 2 prediction to take the form $u_k = -K_{k-n_c}x_k,\ \ k = n_c, ..., n_c + m_c - 1$. The matrix H should of course be chosen such that it implies an increase in the feasibility region. Denote the feasible region for the two last modes as $\mathcal{S}_{E2}$.

The cost function J for the triple mode predictions will (assuming for simplicity that $n_c = m_c$) take the form $J = C^{\mathsf{T}} W_D C + d^{\mathsf{T}} W_D d$. As $d = Hx_{k+n_c}$ and with $\Phi = A - BK$, $x_{k+n_c} = \Phi^{n_c} x_k + [\Phi^{n_c-1}B, ..., B]C$, hence $J = C^{\mathsf{T}} W_T C + C^{\mathsf{T}} V_T x_k + p$, for suitable $W_T,\ V_T$ [6].

The dynamics of the closed loop system with the last two modes can be formulated as an autonomous system, hence a polyhedral invariant set (MAS) can easily be found [16]. Moreover, using this set together with $d = Hx_{k+n_c}$ and $x_{k+n_c} = \Phi^{n_c} x_k + [\Phi^{n_c-1}B, ..., B]C$ it is straightforward to find a MCAS of the form

$$\mathcal{S}_{TE} = \{x : M_{TE} x + N_{TE} C \leq d_{TE}\}. \tag{9}$$

Algorithm 2.2: The LTI Triple mode MPC law, for $x \notin \mathcal{S}_0$, can be summarised as:

$$\min_C J = C^{\mathsf{T}} W_T C + C^{\mathsf{T}} V_T x \text{ s.t. } M_{TE} x + N_{TE} C \leq d_{TE}.$$

It is noted that due to the suboptimal trajectory in the middle mode, the optimal C is not zero, even when $x \in \mathcal{S}_{E2}$.

The advantages of using structure (8) are manyfold:

1) The predictions have a linear dependence on C.
2) The terminal region $\mathcal{S}_{E2}$ may be significantly enlarged over $\mathcal{S}_0$ and hence $\mathcal{S}_{TE}$ maybe much larger than $\mathcal{S}_c$. Hence one has essentially built into the predictions a gradual re-tuning of K [7] as the state moves nearer to the origin.
3) The predictions still retain the 'optimal' feedback asymptotically and this helps to ensure that the performance being minimised is still close to the 'global' optimum.

However, there are weaknesses too. The region $\mathcal{S}_{E2}$ depends strongly upon $K_0 = -K + [I, 0, \ldots, 0]\, H$ as the first implied control action in mode 2 is $u = -K_0 x$. Hence, the terminal region is still restricted to those that can be determined with a fixed state feedback. Another possible weakness is the need for a systematic tool for identifying the best sequence of K_i (that is, finding H). This was addressed in the LTI case in [6, 12], and is reviewed in Section III-C.

Finally, triple mode algorithms utilising polyhedral sets have so far only been developed for the LTI case. The main purpose of this paper is to show how these algorithms can be extended to the robust case.

III. Robust dual mode MPC algorithm

Before we define a robust triple mode algorithm, we recapitulate two dual mode robust MPC algorithms that are instrumental to the triple mode algorithm. The first is based on invariant polyhedrons, while the second is based on invariant ellipsoids.

The class of uncertain systems considered, is linear systems with polytopic uncertainty, that is,

$$x_{k+1} = A(k)x_k + B(k)u_k, \tag{10}$$

where the system matrices are time-varying within matrix polytopes,

$$(A(k), B(k)) \in \Omega := \mathrm{Co}\{(A_1, B_1), \ldots, (A_p, B_p)\}. \tag{11}$$

States and inputs are still subject to constraints (5).

A. Robust dual mode MPC based on invariant polyhedra

One can only extend conventional (QP-based) MPC to the robust case if one can determine suitable robust invariant polyhedral sets; such work was recently given [13, 14] for the standard dual mode Algorithm 2.1. It was shown that, subject to a quadratic stabilisability criterion (also required for ellipsoidal invariance) then the robust MCAS takes the same form as (6), but obviously with different $M_0,\ N_0,\ d_0$. The algorithm works by recursively finding the prior reach sets, and a new invariance condition allows removing redundant constraints at each iteration to avoid combinatorial explosions in the number of terms.

B. Robust dual mode MPC based on ellipsoids

The idea of augmenting the system with the n_c future control degrees of freedom was proposed in [9]. By doing this, feasibility could be handled offline by optimising the size of an invariant ellipsoid subject to constraints and the augmented dynamics. This offline optimisation problem is convex. They denoted this as Efficient Robust Predictive Control (ERPC).

It was noted in [10, 17] that very large ellipsoids could be obtained for smaller n_c if one allowed dynamics in the predictions, at the cost of a non-convex (BMI) offline problem. Shortly after, in [11], it was shown that if $n_c \geq n_x$, it is possible to specify an equivalent *convex* semi-definite programming problem, and moreover, that in terms of size of feasibility region, there is no advantage in choosing $n_c > n_x$. In the sequel, we will refer to this offline problem as *generalised* ERPC (GERPC); the rest of this section will specify the GERPC offline problem (with ERPC as a special case) in more detail.

The GERPC augmented system is

$$z_{k+1} = \begin{bmatrix} A(k)-B(k)K & B(k)D \\ 0 & G \end{bmatrix} z_k;\ z = \begin{bmatrix} x \\ f \end{bmatrix}, \quad (12)$$

where D and G are variables that are used to optimise size and shape of the invariant ellipsoid. ERPC used fixed $D = E$ and $G = I_L$, where

$$E = [I, 0, \ldots, 0], \quad I_L = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ & & \ddots & & \\ 0 & \cdots & 0 & 0 & I \\ 0 & \cdots & 0 & 0 & 0 \end{bmatrix}.$$

In this case, f is the vector of future control perturbations, while in the general (GERPC) case, f defines future control perturbations through the dynamics $u_k = -Kx_k + Df_k$, $f_{k+1} = Gf_k$.

The existence of an ellipsoid $\mathcal{E}_z = \{z \ : \ z^\mathsf{T} Q_z^{-1} z \leq 1\}$ ensuring robust invariance and feasibility (GERPC offline problem) is guaranteed if there exist D, G, Q_z and W such that

$$\star\ Q_z^{-1} \begin{bmatrix} \Phi_j & B_j D \\ 0 & G \end{bmatrix} - Q_z^{-1} < 0, \quad j = 1, \ldots, p, \quad (13a)$$

$$\begin{bmatrix} W & \begin{bmatrix} L_x - L_u K & L_u D \end{bmatrix} \\ \star & Q_z^{-1} \end{bmatrix} > 0, \quad W_{ii} \leq l_i^2, \quad (13b)$$

where $\Phi_j = A_j - B_j K$ (and $\star$ means implied by symmetry). The size of the projection of $\mathcal{E}_z$ to the x-space, $\mathcal{E}_x$ (the feasibility region) is proportional to $\ln\det(TQ_zT^\mathsf{T})$. Hence maximisation of $\mathcal{E}_x$ is obtained by

$$\min_{Q_z, D, G} \ln\det(TQ_zT^\mathsf{T})^{-1} \text{ subject to (13).} \quad (14)$$

It was shown in [11] that there exists convex LMIs that are equivalent to (13), hence this problem can be solved by semidefinite programming packages (for instance, those packed with [18]). Furthermore, it was shown that the cost can be upper bounded by γ if (13a) is modified to

$$\begin{aligned} \star\ Q_z^{-1} \begin{bmatrix} \Phi_j & B_j D \\ 0 & G \end{bmatrix} - Q_z^{-1} \\ < -\frac{1}{\gamma} \star \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} I & 0 \\ -K & D \end{bmatrix}, \quad j = 1, \ldots, p. \end{aligned} \quad (15)$$

This γ provides a direct tuning parameter for the size of feasibility region vs. online cost trade-off for GERPC. The reader is referred to [11] for more details.

C. Using (G)ERPC for nominal triple mode

One approach for finding the H in (8) (LTI triple mode) is via the LTI version of the (G)ERPC offline problem; this was proposed in [6] for ERPC and in [12] for GERPC.

Write the invariant set $\mathcal{E}_z$ in $z = \left[x^\mathsf{T}, f^\mathsf{T}\right]^\mathsf{T}$-space as

$$[x^\mathsf{T}, f^\mathsf{T}] \underbrace{\begin{bmatrix} P_{11} & P_{12} \\ P_{12}^\mathsf{T} & P_{22} \end{bmatrix}}_{Q_z^{-1}} \begin{bmatrix} x \\ f \end{bmatrix} \leq 1. \quad (16)$$

Now, the projection onto x-space is given by

$$\mathcal{E}_x = \{x : \exists f \text{ s.t. } x^\mathsf{T} P_{11} x \leq 1 - f^\mathsf{T} P_{22} f - 2x^\mathsf{T} P_{12}^\mathsf{T} f\}.$$

The inequality can be rewritten as

$$x^\mathsf{T}[P_{11} + H^\mathsf{T} P_{22} H]x \leq 1 - (f - Hx)^\mathsf{T} P_{22}(f - Hx),$$

where $H = -P_{22}^{-1} P_{21}$. It is easy to see that the size of $\mathcal{E}_x$ is maximised when $f = Hx$, meaning that the maximal feasible region is given by a fixed linear state feedback. This f takes the form required by the triple mode algorithm of section II-B ($d = f$) and hence the corresponding H is a possible choice, as first suggested in [6].

For GERPC, rather than setting $d = Hx_{k+n_c}$ (which is possible), it is a better idea to let d_i develop through the GERPC dynamics (12), that is,

$$\begin{aligned} d_0 &= DHx, \\ d_1 &= DGHx, \\ d_2 &= DG^2Hx, \\ &\vdots \end{aligned} \quad (17)$$

as suggested in [12]. For ERPC, these two approaches turn out to be exactly the same due to the structure of I_L. But since G is not nil-potent in general, GERPC does not have a finite horizon impact on the input, rather it defines an infinite sequence of control moves.

Remark 3.1: It is tempting to use the same approach for triple mode *robust* MPC, that is, using H which just as in the LTI case can be shown to ensure robust invariance of the (G)ERPC ellipsoid. However, since the implied control

after the first mode will depend on the (in the robust case) uncertain x_{k+n_c} through $d_0 = DHx_{k+n_c}$, it is hard to establish the recursive feasibility required in "standard" proofs of stability of robust MPC laws [2,19]. Hence, a slight twist to the triple mode setup is needed, provided in the next section.

Remark 3.2: Since a state feedback ($u_k = (-K + DH)x_k$) can be found that has the same (robustly) feasible region as (G)ERPC, a dual mode robust MPC algorithm can be constructed by the methods in [20] using this state feedback as terminal control (defining terminal set and cost). Although this will give a large feasibility region (size vs. cost to a degree tunable with γ), online cost will be suboptimal since the cost (for small n_c) will be dominated by the (G)ERPC state feedback designed for maximum feasibility. However, suboptimality can be reduced by switching between several dual mode algorithms based on decreasing γs.

IV. Robust triple mode MPC

This section will show how the solutions of the robust (G)ERPC offline problem can be used to specify a robust triple mode MPC algorithm deploying polyhedral sets.

A. Constraints for triple mode

The triple mode setup (predicted controls) will be slightly different than for the nominal case (8):

$$\begin{array}{ll} u_k = -Kx_k + d_k + c_k, & k = 0, \ldots, n_c - 1, \\ u_k = -Kx_k + d_k, & k = n_c, \ldots, n_c + m_c - 1, \\ u_k = -Kx_k & k \geq n_c + m_c. \end{array} \quad (18)$$

The c_k will be the degrees of freedom for the QP-based triple mode algorithm, while the d_k will be defined from the (G)ERPC offline solution as $d_k = Df_k$, $f_{k+1} = Gf_k$. It is noteworthy that for non-nilpotent G, $m_c = \infty$ and thus the third mode only asymptotically enters the predictions.

Augmenting the (already augmented) dynamics (12) with $C_k = \left[c_0^\mathsf{T}, c_1^\mathsf{T}, \ldots, c_{n_c-1}^\mathsf{T}\right]^\mathsf{T}$, the following autonomous system describes the predicted dynamics:

$$\begin{bmatrix} x_{k+1} \\ f_{k+1} \\ C_{k+1} \end{bmatrix} = \begin{bmatrix} A(k) - B(k)K & B(k)D & B(k)E \\ 0 & G & 0 \\ 0 & 0 & I_L \end{bmatrix} \begin{bmatrix} x_k \\ f_k \\ C_k \end{bmatrix}. \quad (19)$$

These dynamics should fulfill the constraints (5),

$$\begin{bmatrix} L_x - L_u K & L_u D & L_u E \end{bmatrix} \leq l. \quad (20)$$

Lemma 4.1: A robustly invariant polyhedral set can be found for (19) subject to (20) using the algorithm described in Section III-A, on the form

$$Mx + N_1 f + N_2 C \leq d. \quad (21)$$

Proof omitted as obvious. Denote the polyhedron $\mathcal{S}_{aug}$.

Define the vertices in a polytopic uncertainty description of (19) as

$$\Psi_i := \begin{bmatrix} A_i - B_i K & B_i D & B_i E \\ 0 & G & 0 \\ 0 & 0 & I_L \end{bmatrix}.$$

We see that the number of vertices is the same as for (10).

It is clear that for a "first mode" control sequence C, the autonomous system (19) captures future state behavior from initial conditions x_0 and f_0.

B. Costs for triple mode

A cost $J(x, f, C)$ that bounds the infinite horizon cost should be found:

$$J(x, f, C) \geq \max_{(A(i),B(i)) \in \Omega} \sum_{k=0}^{\infty} x_k^\mathsf{T} Q x_k + u_k^\mathsf{T} R u_k. \quad (22)$$

Such a cost can be constructed as

$$J(x, f, C) = \left[x \; f \; C\right]^\mathsf{T} P \left[x \; f \; C\right], \quad (23)$$

with $P > 0$ satisfying

$$\begin{aligned} P - \Psi_i^\mathsf{T} P \Psi_i \geq & \left[I \; 0 \; 0\right]^\mathsf{T} Q \left[I \; 0 \; 0\right] \\ & + \left[-K \; D \; E\right]^\mathsf{T} R \left[-K \; D \; E\right], \; i = 1, \ldots, p. \end{aligned} \quad (24)$$

The matrix P can be efficiently calculated by the SDP

$$\min \operatorname{tr}(P) \text{ subject to (24).} \quad (25)$$

Lemma 4.2: The cost (23) satisfies (22).

Proof: Since (24) is convex in Ψ_i, it implies that

$$\begin{aligned} P - \Psi(k)^\mathsf{T} P \Psi(k) \geq & \left[I \; 0 \; 0\right]^\mathsf{T} Q \left[I \; 0 \; 0\right] \\ & + \left[-K \; D \; E\right]^\mathsf{T} R \left[-K \; D \; E\right]. \end{aligned}$$

Multiplying with augmented state from both sides and summation over $k = 0, \ldots, \infty$ gives the result. ∎

C. Triple mode algorithm and stability

Algorithm 4.1 (Triple mode MPC): Given design parameters n_c, Q, R, and γ, calculate D and G from (14), and P from (25).

1) $k = 0$: Given x_0, calculate a feasible f_0 from (21). This is an LP, or an QP if the minimum norm f_0 is desired.
2) Solve the optimisation problem

   $$\min_C J(x_k, f_k, C) \text{ subject to } (x_k^\mathsf{T}, f_k^\mathsf{T}, C^\mathsf{T})^\mathsf{T} \in \mathcal{S}_{\text{aug}}.$$

   As J is quadratic and $\mathcal{S}_{aug}$ is polyhedral, this is an QP. Denote the optimal C with $C^\star$.
3) Implement $u_k = -Kx_k + Df_k + EC^\star$ to the plant. Calculate $f_{k+1} = Gf_k$.
4) Set $k = k + 1$, go to 2).

Proving stability follows the well-known path [2] of first showing recursive feasibility, and then showing that the cost function serves as a Lyapunov function.

Lemma 4.3 (Recursive feasibility): If Algorithm 4.1 is feasible at $k = 0$, then it is feasible for $k > 0$.

Proof: Given an optimal $C^\star = \left[c_0^\mathsf{T}, c_1^\mathsf{T}, \ldots, c_{n_c-1}^\mathsf{T}\right]^\mathsf{T}$ at time k, it is clear from (19) and invariance of $\mathcal{S}_{aug}$ that $C = \left[c_1^\mathsf{T}, c_2^\mathsf{T}, \ldots, c_{n_c-1}^\mathsf{T}, 0\right]^\mathsf{T}$ is a feasible first mode control sequence at time $k + 1$. Repeating this argument proves the lemma. ∎

Theorem 4.1 (Robust asymptotic stability): Algorithm 4.1 robustly asymptotically stabilises system (10) subject to constraints (5).

Proof: Feasibility at each time step follows from Lemma 4.3. Asymptotic stability follows since $J^\star(k) := J(x_k, f_k, C_k^\star)$ is positive definite and monotonically increasing (is a Lyapunov function): Using the optimal input at

time k to construct a feasible input for time $k+1$ as in the proof of Lemma 4.3, it is clear from Lemma 4.2 that $J^\star(k+1) \leq J(x_{k+1}, f_{k+1}, \left[c_1^\mathsf{T}, c_2^\mathsf{T}, \ldots, c_{n_c-1}^\mathsf{T}, 0\right]^\mathsf{T}) < J^\star(k)$ for $x(k) \neq 0$. ■

D. Comments regarding the triple mode algorithm

The size of the GERPC ellipsoid (and hence the feasibility region of Algorithm 4.1) can be tuned against online performance with the parameter γ. For GERPC, the dimension of f should be chosen equal to n_x and not be used for tuning, since if $\dim(f) > n_x$ does not give larger ellipsoids and $\dim(f) < n_x$ gives a non-convex offline problem. For ERPC one has to use $\dim(f)$ to tune the size of the ellipsoid, but the freedom for tuning is much less, since the ERPC ellipsoids tends to increase most in certain directions, and the offline problem becomes intractable for large $\dim(f)$.

The augmented state f of the triple mode algorithm must be initialised and simulated along with the real system. This differs from the nominal triple mode algorithm (Section III-C), and implies that the robustly invariant polyhedra must be found for a higher dimensional system than in the nominal case. However, the increase in dimension does not give more vertices in the uncertainty description.

Although the necessity of this online augmented state will be investigated closer, it has some significant advantages:

- If Algorithm 4.1 at any point should become infeasible (due to an unmodeled disturbance), the algorithm can be reset by calculating a new f_0.
- The state f can be reset (minimised as in step 1 in Algorithm 4.1) at certain time-steps to improve performance, since a smaller f means the triple mode algorithm is closer to the corresponding dual mode.

The convergence of f can be seen as a gradual tuning of the MPC algorithm as the state approaches the origin. For $f = 0$, the algorithm reduces to the ordinary dual mode algorithm.

V. Example

Consider the system described by

$$(A_1, B_1) = (\left[\begin{smallmatrix} 1 & 0.1 \\ 0 & 1 \end{smallmatrix}\right], \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]),\ (A_2, B_2) = (\left[\begin{smallmatrix} 1 & 0.2 \\ 0 & 1 \end{smallmatrix}\right], \left[\begin{smallmatrix} 0 \\ 1.5 \end{smallmatrix}\right]),$$
$$-1 \leq u_k \leq 1, \quad \begin{bmatrix} -10 & -10 \end{bmatrix}^\mathsf{T} \leq x_k \leq \begin{bmatrix} 10 & 10 \end{bmatrix}^\mathsf{T}.$$

and cost defined by $Q = I$ and $R = 0.01$. The third mode control (the second mode for dual mode MPC) K is given as the unconstrained LQ optimal of the average of the vertices in the uncertainty description. Furthermore, for triple mode we choose $n_c = 2$ (second mode) and $\dim(f) = n_x = 2$.

Figure 1 shows a comparison of feasibility regions for robust dual mode based on [13,20] (with $n_c = 2$ and $n_c = 4$), and Algorithm 4.1 with G, D calculated based on GERPC for $\gamma = 10$ and $\gamma = 1e4$. It is clear that including a triple mode substantially increases the feasibility region, and larger γ increases it further.

This increase comes at the price of an increase in number of constraints in the online QP problem, as can be seen in the first row of Table I (note that dual mode with $n_c = 4$ have four QP optimisation variables, while the others have two). However, calculating the polyhedra using the algorithm in [21] reduces the number of constraints to a comparable level to dual mode, with a very small decrease in volume of feasibility region (not shown here).

	Dual		Triple	
	$n_c = 2$	$n_c = 4$	$\gamma = 10$	$\gamma = 1e4$
# constraints [13]	18	66	144	118
# constraints [21]	18	38	58	52

TABLE I: Number of constraints in online QP for dual and triple mode MPC. The first row is based on finding polyhedral regions using the algorithm in [13, 20], while the second is based on [21].

An online cost comparison would require the same initial conditions. The minimisation of f_0 in the first step in Algorithm 4.1 ensures that $f_0 = 0$ (or very small) for (the small) initial conditions that are feasible for the dual mode algorithm. This would mean that triple mode reduces to dual mode, with essentially the same cost.

VI. Conclusion and future works

Robust MPC based on a triple mode prediction setup allows a QP-based MPC algorithm with large feasibility regions for a small number of online optimisation variables. The price to pay is a limited increase in number of constraints online, and a larger offline problem than dual mode counterparts. A tuning parameter in the offline problem allows increasing the feasibility region to a very large extent, and a minor modification to the online algorithm ensures that this has little effect on overall online performance.

Since the robust triple mode MPC algorithm is based on QPs, it is amenable to explicit solutions based on multiparametric QPs.

VII. Acknowledgments

Lars Imsland acknowledges support from the Research Council of Norway through the BIGCO2 project. Bert Pluymers' research partially supported by KUL: GOA-Mefisto 666, GOA-AmbioRics; FWO: G.0240.99, G.0407.02, G.0197.02, G.0141.03, G.0491.03, G.0120.03, G.0800.01, G.0452.04, G.0499.04, G.0211.05, G.0080.01, G.0226.06, research communities (ICCoS, ANMMM); IWT: PhD Grants, BFSPO: IUAP P5/22; PODO-II (CP/40: TMS and Sustainability); EU: FP5-CAGE; FP5-Quprodis; ERNSI; FP6-BioPattern; Eureka 2419-FliTE; Contract Research/agreements: Data4s, Electrabel, Elia, LMS, IPCOS, VIB; Johan Suykens' research is partially funded by : GOA-Ambiorics, IAP V, FWO projects G.0407.02, G.0211.05, G.0499.04, G.0226.06.

References

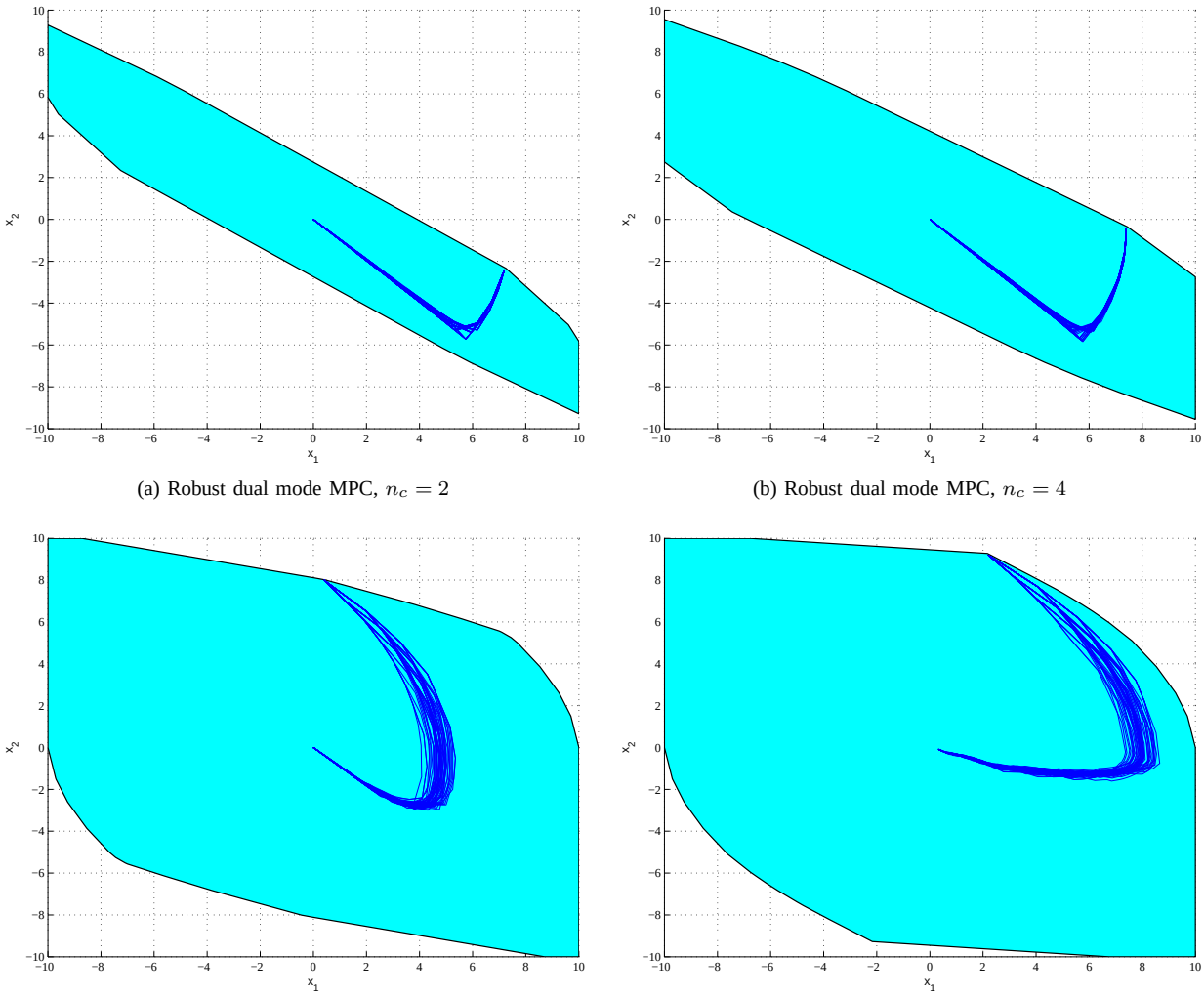
[1] J. A. Rossiter, *Model-based Predictive Control - a Practical Approach*. CRC Press, 2003.

[2] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[3] P. O. M. Scokaert and J. B. Rawlings, "Constrained linear quadratic regulation," *IEEE Trans. Aut. Control*, vol. 43, no. 8, pp. 1163–1169, 1998.

[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*, vol. 38, pp. 1061–1068, 2002.

[5] J. A. Rossiter, B. Kouvaritakis, and M. Bacic, "Interpolation based computationally efficient predictive control," *Internat. J. Control*, vol. 77, no. 3, pp. 290–301, 2004.



(a) Robust dual mode MPC, $n_c = 2$

(b) Robust dual mode MPC, $n_c = 4$

(c) Robust triple mode MPC, $n_c = 2$, $\gamma = 10$

(d) Robust triple mode MPC, $n_c = 2$, $\gamma = 1e4$

Fig. 1: Feasibility regions for dual and triple mode MPC. 50 closed loop trajectories plotted for each MPC control law.

[6] J. A. Rossiter, B. Kouvaritakis, and M. Cannon, "Computationally efficient algorithms for constraint handling with guaranteed stability and near optimality," *Internat. J. Control*, vol. 74, no. 17, pp. 1678–1689, 2001.

[7] K. T. Tan and E. G. Gilbert, "Multi-mode controllers for linear discrete-time systems with general state and control constraints," in *Optimization: Techniques and applications, Vol. 1, 2 (Singapore, 1992)*. River Edge, NJ: World Sci. Publishing, 1992, pp. 433–442.

[8] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, January 2002.

[9] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans, "Efficient robust predictive control," *IEEE Trans. Aut. Control*, vol. 45, no. 8, pp. 1545–1549, 2000.

[10] L. Imsland, N. Bar, and B. A. Foss, "More efficient predictive control," *Automatica*, vol. 41, no. 8, 2005.

[11] M. Cannon and B. Kouvaritakis, "Optimizing prediction dynamics for robust MPC," *IEEE Trans. Aut. Control*, vol. 50, no. 11, pp. 1892–1897, 2005.

[12] L. Imsland and J. A. Rossiter, "Time varying terminal control," in *Proc. 16th IFAC World Congress, Prague, Czech Republic*, 2005.

[13] B. Pluymers, J. Rossiter, J. Suykens, and B. De Moor, "The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty," in *Proc. Amer. Contr. Conf.*, 2005.

[14] B. Pluymers, J. Rossiter, J. Suykens, and B. D. Moor, "A simple algorithm for robust MPC," in *Proc. 16th IFAC World Congress, Prague, Czech Republic*, 2005.

[15] J. A. Rossiter, B. Kouvaritakis, and M. J. Rice, "A numerically robust state-space approach to stable-predictive control strategies," *Automatica*, vol. 34, no. 1, pp. 65–73, 1998.

[16] 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*, vol. 36, no. 9, pp. 1008–1020, 1991.

[17] S. Drageset, L. Imsland, and B. A. Foss, "Efficient model predictive control with prediction dynamics," in *Proc. 7th European Control Conference ECC'03*, Cambridge, England, 2003.

[18] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004, available from `http://control.ee.ethz.ch/~joloef/yalmip.php`.

[19] B. Pluymers, J. A. K. Suykens, and B. de Moor, "Min-max feedback MPC using a time-varying terminal constraint set and comments on "Efficient robust constrained model predictive control with a time-varying terminal constraint set"," *Systems Control Lett.*, vol. 54, no. 12, pp. 1143–1148, 2005.

[20] B. Pluymers, J. A. Rossiter, J. A. K. Suykens, and B. De Moor, "The efficient computation of polyhedral invariant sets for LPV systems and application to robust MPC," 2005, submitted for journal publication.

[21] B. Pluymers, J. A. K. Suykens, and B. de Moor, "Construction of reduced complexity polyhedral invariant sets for LPV systems using linear programming," *Automatica*, 2005, submitted for publication.