2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

The impact of the input parameterisation on the feasibility of MPC and its parametric solution

J.A.Rossiter and S.S. Dughman

Abstract— **Feasibility is an important issue in predictive control, but the influence of many important parameters such as the desired steady-state, or target, the current value of the input are rarely discussed in the literature. This paper makes two contributions. First it gives visibility to the issue that including core parameters such as the target and the current input vastly increases the dimension of the parametric space, with possible consequences on the complexity of any parametric solutions. Secondly, it is shown that a simple re-parameterisation of the d.o.f. to take advantage of reference governor concepts can lead to large increases in feasible volumes, with no increases in the dimension of the required optimisation variables.**

I. Introduction

A key selling point of predictive control [4],[12] is the ability to handle constraints, systematically, in the design of an optimum control strategy. This enables operators to push systems closer to their limits and thus improve productivity and/or quality. However, a little discussed consequence of including constraints is the so called *feasibility problem*. Hereafter, feasibility means that the *class of predictions* over which an optimisation is being performed, includes at least one which is able to satisfy all the constraints. Infeasibility means the class of predictions does not include a selection which satisfies all constraints, hence the predictive control law is undefined with a number of undesirable consequences.

It is of interest to consider the extent to which infeasible MPC optimisations can occur and also, what mitigating action might be appropriate. This issue was recognised early on in the literature and the simplest solutions provided were so called *reference governors* [7], [1]. In these, a tacit assumption is made that infeasibility is often caused by a fast change of the set point and thus, feasibility can be retained by slowing down set point changes. In the early literature, the algorithms focussed on simple computations for determining the required slow-down; optimality was not a main criteria. An alternative approach in the standard MPC literature [16], [11], [19], [15], [21], is to augment the degrees of freedom with a target deviation term, that is, to allow the target to deviate from its true value so that the associated predictions are feasible. Of course, the downside of such an approach is that one may end up optimising performance with respect to the incorrect target, although in truth, such an issue seems somewhat hypothetical if the true target is unreachable.

Typically, authors have used the standard parameterisation of the degrees of freedom (d.o.f.), that is deviations to the first few control moves [22], [17]. More recently, a few authors have looked at the potential of alternative parameterisations such as those supported by Laguerre polynomials [8], [20], as this embeds a form of slowing down of the transition from one steady-state to another and thus enable fewer d.o.f. to be utilised to capture a whole transient behaviour. However, such approaches have not yet been extended to consider scenarios with a permanent offset from the target and thus extensions to this approach are deferred until later.

Another relatively recent development is the potential to form parametric solutions [13], [2]. These can be advantageous for systems which require fast sample rates and/or some form of rigorous validation of closed-loop behaviour. The downside is that some times the parametric solutions are very complex, requiring excessive offline computation and excessive online storage and set membership tests [9]. Thus, there is interest if finding alternatives which give simple solutions, perhaps at the cost of some suboptimality [3].

This paper investigates the two issues discussed above, that is to investigate the potential of different input parameterisations to reduce the complexity of parametric solutions alongside a simple aim of achieving large enough feasible regions [18]. Nevertheless, rigorous analytic results are not possible due to the highly nonlinear interdependence between feasible regions, constraints, degrees of freedom and the underlying system. Instead, here we take a case study based approach to explore alternatives using the argument that, if different solutions have significant benefits for some systems, they may also do so for other systems and thus this is useful knowledge; this paper focuses on square systems only as non-square systems have a number of additional issues which complicate the scenario further. The focus is on dual-mode approaches to MPC as these have straightforward guarantees of closed-loop stability whereas finite horizon approaches do not. The 2nd section introduces basic background material on dual-mode MPC and also, gives explicit details of how integral action, tracking and steady-state offset are incorporated into the problem (such details are often avoided in the literature). Section 4 presents several examples of how feasibility varies for different parameterisations of the d.o.f. Section 5 focuses on parametric solutions to explore how complexity of these might, or might not, be linked to the choice of parameterisation. Conclusions are in section 6.

II. Background on dual-mode predictive control

This section summarises the optimal predictive control (OMPC) [17], [22] algorithm with specific attention given to how the problem is augmented to include reference

Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD. j.a.rossiter@sheffield.ac.uk
S. Dughman is sponsored by the Libyan Government.
ssdughman@sheffield.ac.uk

trajectories and steady-state offsets. Such detail is important to understand the dimension of the associated parametric space, as extra states are required to include: (i) integral action/targets; (ii) definition of input rates; (iii) steady-state offsets. The need for each of these states is often tacitly ignored but, this increase in dimension has significant repercussions for computational loading and data storage.

Remark 1: Proofs of recursive feasibility and stability are established in the literature and thus not repeated as this paper contribution focusses on the impact of different choices on the volumes of feasible regions.

A. Nominal model

For simplicity this paper assumes an observable and controllable state space model of the following form.

$$x_{k+1} = Ax_k + Bu_k; \quad y_k = Cx_k + d_k \tag{1}$$

with x_k, u_k, y_k, d_k the states, input, output and disturbance respectively with dimensions n_x, m, m, m and the input rate defined from $\Delta u_k = u_k - u_{k-1}$. The disturbance signal d_k, which is estimated, is used to incorporate integral action and cope with parameter uncertainty. The system is subject to constraints, typically (others are possible):

$$\underline{u} \le u_k \le \overline{u}; \quad \underline{\Delta u} \le \Delta u_k \le \overline{\Delta u}; \quad \underline{x} \le K_x x_k \le \overline{x} \tag{2}$$

Define the future target $\underset{\rightarrow k+1}{r}$ as (assumed constant):

$$\underset{\rightarrow k+1}{r} = [r_{k+1}^T, r_{k+2}^T, \cdots, r_{k+n_y}^T]^T = [I, I, \ldots]^T r_{k+1} \tag{3}$$

The system steady-state states and inputs are estimated from solving $x_{k+1} = x_k,\ y_k = r_{k+1}$ and hence:

$$\begin{bmatrix} A-I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_{ss} \\ u_{ss} \end{bmatrix} = \begin{bmatrix} 0 \\ r_{k+1} - d_k \end{bmatrix} \tag{4}$$

$$\rightarrow \quad \begin{bmatrix} x_{ss} \\ u_{ss} \end{bmatrix} = \begin{bmatrix} K_{xr} \\ K_{ur} \end{bmatrix} (r_{k+1} - d_k)$$

B. Optimal or dual-mode predictive control

It is now well known [12] that dual-mode approaches have good stability properties in general. Here, a standard OMPC algorithm is defined. OMPC uses a infinite horizon performance index of the following form:

$$J = \sum_{i=0}^{\infty} (x_{k+1+i} - x_{ss})^T Q (x_{k+1+i} - x_{ss}) + (u_{k+i} - u_{ss})^T R (u_{k+i} - u_{ss}) \tag{5}$$

Define $u_k - u_{ss} = -K(x_k - x_{ss})$ to be the optimal unconstrained feedback minimising (5); this can be solved with standard identities. A key selling point of OMPC is the ability to handle constraints. To do this two things are needed; (i) a definition of the system predictions so these can be compared to the constraints (2) for all future samples and (ii) some d.o.f. which can be used to vary the predictions about the nominal unconstrained optimal predictions arising from the use of feedback $u_k - u_{ss} = -K(x_k - x_{ss})$.

C. Input parameterisation and modified performance index

A common input parameterisation takes the form:

$$\begin{array}{ll} u_{k+i} - u_{ss} = -K(x_{k+i} - x_{ss}) + c_{k+i} & i = 0, 1, \cdots, n_c - 1 \\ u_{k+i} - u_{ss} = -K(x_{k+i} - x_{ss}) + c_{\infty} & i \ge n_c \end{array} \tag{6}$$

so the variables $c_{k+i}, i = 0, 1, \ldots, n_c - 1$ are the degrees of freedom which allow deviations in the first n_c moves of the optimal input trajectory; the term c_∞ is a d.o.f. which enables steady-state offset between the asymptotic output predictions and desired target r_{k+1}. Substituting (6) into (5), it is well known [17] that minimising the performance index wrt to $\underset{\rightarrow k}{c}$ is equivalent to minimising the following (with $S_D = diag(S, S, \ldots)$):

$$J = \underset{\rightarrow k}{c}^T S_D \underset{\rightarrow k}{c} + \sum_{\infty} c_\infty^T S c_\infty; \ \ \underset{\rightarrow k}{c} = [c_k^T, \cdots, c_{k+n_c-1}^T]^T \tag{7}$$

Remark 2: Given the sum to infinity of error terms, J of (7) is minimised by minimising the term $c_\infty^T S c_\infty$, that is, unsurprisingly, minimising the offset. One might argue that choosing $c_\infty = 0$ is a reasonable choice, if possible. However, putting all the focus on the asymptotic offset can be disadvantageous to transient errors, so in practice a trade off can be achieved using something along the lines of:

$$J = \underset{\rightarrow k}{c}^T S_D \underset{\rightarrow k}{c} + \lambda c_\infty^T S c_\infty \tag{8}$$

where λ is a weighting to be selected.

D. Predictions

Formulating the predictions needs a little more care because the predictions are used to ensure the expected behaviour satisfies constraints. In consequence, due to the implied closed-loop form of (6), the predictions must include information such as the future target, measured disturbance and current input. A convenient means of combining (1,6) is with an autonomous model formulation [10]. In this case, the formulation must be extended to capture the evolution of $r_{k+1} - d_k, x_k, u_k, \Delta u_k$ as these values appear in the constraints (2). Hence, the following identities are needed:

$$\{c_{k+i} = c_\infty, i \ge n_c\} \quad \{r_{k+i} - d_{k+i} = r_{k+1} - d_k, i \ge 1\} \tag{9}$$
$$\{\Delta u_{k+i} = u_{k+i} - u_{k+i-1}, i \ge 0\}$$

Combining (1,4,6,9) gives the following:

$$Z_k = \begin{bmatrix} x_k \\ r_{k+1} - d_k \\ \underset{\rightarrow k}{c} \\ c_\infty \\ u_{k-1} \end{bmatrix}; \tag{10}$$

$$Z_{k+1} = \underbrace{\begin{bmatrix} \Phi & (I-\Phi)K_{xr} & [B, 0, \ldots, 0] & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I_L & I & 0 \\ 0 & 0 & 0 & I & 0 \\ K & -K.K_{xr} - K_{ur} & [I, 0, 0, \ldots] & 0 & 0 \end{bmatrix}}_{\Psi} Z_k$$

where I_L is a block upper triangular matrix of identities.

E. Constraints

The final building block in an MPC algorithm is the set of inequalities which ensure the predictions from model (10) satisfy constraints (2). There are several algorithms for formulating these inequalities (e.g. [6], [14]) so here just the result is given. Combining model (10) and constraints (2), inequalities representing constraint satisfaction of the predictions, for suitable N, T, M, P, Q, f, reduce to:

$$N \underset{\rightarrow k}{c} + Tc_\infty + Mx_k + Pu_{k-1} + Q(r_{k+1} - d_k) \leq f \quad (11)$$

where details of how to compute the parameters are excluded as standard in the admissible set literature [6].

Lemma 1: The constraint inequalities (11) can be expressed in parametric form as (f is assumed constant):

$$\underbrace{[N, T]}_{N_T} \begin{bmatrix} \underset{\rightarrow k}{c} \\ c_\infty \end{bmatrix} + \underbrace{[M, P, Q]}_{M_w} W_k \leq f; \quad W_k = \begin{bmatrix} x_k \\ u_{k-1} \\ r_{k+1} - d_k \end{bmatrix} \quad (12)$$

where the parameter space is W_k and the d.of. are in $\underset{\rightarrow}{c}, c_\infty$.

Theorem 1: Including both tracking and input rate constraints into OMPC increases the effective parameter space by the dimensions of the target r_k and the input u_k respectively compared to the scenarios where these are excluded.

Proof: Self-evident from the definition of W_k in (12). □

This latter point is important because, it highlights a little discussed impediment to the widespread adoption of parametric approaches. It is well accepted that parametric approaches suffer from computational challenges as the state dimension increases [2], [9], but it is rarely highlighted that it is not just the state dimension which is an issue, but also the significant increase in the implied state dimension required to include input rate constraints and offset free tracking.

Corollary 1: As one aspect of this paper is focussed around parametric approaches, future target information has not been included, that is we assume $r_{k+i} = r_{k+1}, \forall i > 0$. To do otherwise would increase the parametric dimension of W_k further still [5].

F. The OMPC algorithm

Having constructed all the foundation components, an OMPC algorithm [22] can now be defined.

Algorithm 1: OMPC is defined as follows. At each sample, perform the quadratic programming optimisation

$$\min_{\underset{\rightarrow k}{c}, c_\infty} \underset{\rightarrow k}{c}^T S_D \underset{\rightarrow k}{c} + \lambda c_\infty^T S c_\infty \quad \text{s.t.} \quad N_T \begin{bmatrix} \underset{\rightarrow k}{c} \\ c_\infty \end{bmatrix} + M_x W_k \leq f; \quad (13)$$

Implement the first value of c_k in (6) to determine the current input, that is u_k.

Remark 3: Strictly speaking the classical OMPC algorithm uses $\lambda = c_\infty = 0$ but this paper includes the extra d.o.f. because the intention is to consider the efficacy of this for simplifying overall complexity and computational load.

G. Summary

This section has defined the core components in an OMPC algorithm which allows for steady-state offset in the predictions, that is an appropriate performance index and also inequalities to capture the constraints. This offset may be used *optionally* as a mechanism to avoid infeasibility in transients [7], even where steady-state feasibility is assured. Moreover, the OMPC framework has been deliberated cast in a format suitable for parametric approaches as these results can now used to investigate two related but separate issues.

1) The extent to which the parameter c_∞ is more or less effective than $\underset{\rightarrow k}{c}$ in increasing the feasible space.
2) The extent to which the parameter c_∞ may or may not simplify parametric solutions as compared to the use of $\underset{\rightarrow k}{c}$ in the cases where the problem includes tracking.

III. Exploiting offset in system predictions

A key point in this paper is how to deal with transient infeasibility in a computationally efficient manner but it is assumed that the asymptotic steady state is feasible (and thus differs from [15]). A key observation is that the class of input predictions (6), with $c_\infty = 0$, may not have sufficient d.o.f. to satisfy constraints (2 or 12) when n_c is small (as is common to ensure the corresponding QP of (13) is manageable). A historical proposal, where the infeasibility was due to a rapid change in the target r_k, was to deploy a reference governor which, in effect, changed the target temporarily. Here, the intention is to deploy the d.o.f. c_∞ which has the same effect, that is, its inclusion is equivalent to a change in the steady-state target. The preference for using c_∞ is that it fits neatly into the prediction structure of (6) and autonomous model of (10) and thus is straightforward to use when assessing feasible regions using admissible sets and inequalities.

A. Using c_∞ to enlarge feasible regions

The first objective is to assess whether adding the d.o.f. c_∞ is more effective than increasing n_c by one; both these changes increase the overall d.o.f. and thus optimisation dimension by the same amount. Any insights gained are useful as, in practice, operators like to keep the overall optimisation dimension as small as reasonably possible.

The concept of n-step sets is widely understood in the MPC literature. In essence:

- A 0-step set is the region in which the control law (6) satisfies constraints when $n_c = 0$ and $c_\infty = 0$. This is where the unconstrained control law is feasible.
- A 1-step set gives the range of values of W_k such that, with a single non-zero value of c_k, one can satisfy constraints at the first sample, and move into the 0-step set by the next sample.
- A 2-step set gives the range of values of W_k such that, with a single non-zero value of c_k, one can satisfy constraints at the first sample, and move into the 1-step set by the next sample.
- The definition of a n-step set follows the same pattern.

Lemma 2: With $c_\infty = 0$ and a given choice of n_c, the feasible region is given by the n_c-step set. This is obvious.

Remark 4: Problems occur when the current states x_k, u_{k-1} are good distance from the target steady-state x_{ss}, u_{ss}. In this case infeasibility can arise as the n_c-step set around the target steady-state is limited in volume, so points far away are not inside if n_c is small. To retain feasibility, it is necessary to choose an alternative n_c-step set, that is one associated to a different W_k; this means change the only component in W_k which you can which is r_k.

Theorem 2: Where a simple move of the implied steady-state x_{ss}, u_{ss} is sufficient to retain feasibility, then the d.o.f. c_∞ will be sufficient to retain feasibility.

Proof: This is obvious as choices for c_∞ exist which can be used to imply convergence to any asymptotically stable steady-state point. □

Theorem 3: Assuming that the OMPC problem was feasible at sample $k-1$, then the inclusion of c_∞ guarantees feasibility at the sample k.

Proof: The main difference between sample k and $k-1$ in terms of the implied predictions in (12), is the change in the value $r_{k+1} - d_k$. It has been shown that c_∞ can overwrite any impact on predictions from a change in that state, and thus can be used to place the system in the same effective state as at the previous sample. □

Remark 5: It is noted that reachable steady-states are limited to the sub-space implicit in (4). Where a simple move of the implied steady-state x_{ss}, u_{ss} is not sufficient to retain feasibility, then the d.o.f. c_∞ is less likely to be useful and hence one get more benefit from increasing n_c. This case will occur where the initial condition rather than changes in the target cause infeasibility.

IV. Numerical examples

This section will show how the shapes of the feasible regions vary for changes in n_c and the inclusion or not of c_∞. For ease of illustration examples are restricted to a parameter space of dimension 2. One example is:

$$A = \begin{bmatrix} 0.8 & 0.1 \\ -0.2 & 0.9 \end{bmatrix}; \; B = \begin{bmatrix} 0.3 \\ 0.8 \end{bmatrix}; \; -0.2 \le u \le 0.5; \quad (14)$$

$$\|\Delta u_k\| \le 0.05; \begin{bmatrix} 1 & 0.2 \\ -0.1 & 0.4 \\ -1 & 0.2 \\ 0.1 & -0.4 \end{bmatrix} x_k \le \begin{bmatrix} 8 \\ 8 \\ 1.6 \\ 5 \end{bmatrix}$$

A 2nd example has:

$$A = \begin{bmatrix} 0.8 & -0.53 \\ -0.09 & 0.97 \end{bmatrix}; \; B = \begin{bmatrix} 0.09 \\ 0.005 \end{bmatrix}; \; -5 \le u \le 4; \quad (15)$$

$$\|\Delta u_k\| \le 0.4; \begin{bmatrix} 1 & 0.2 \\ -0.1 & 0.4 \\ -1 & -0.2 \\ 0.1 & -0.4 \\ -1 & -0.45 \end{bmatrix} x_k \le \begin{bmatrix} 4 \\ 1.6 \\ 0.8 \\ 1.6 \\ 0.6 \end{bmatrix}$$

A. With and without c_∞ but varying u_{k-1}

A little discussed issue in the literature is the impact of the initial input on the feasible regions; this is relevant when there are input rate constraints and it is also clear that u_{k-1} is one component of the parametric space W_k.

- Figures 1,3 show how the 2-step set for examples (14,15) changes as u_{k-1} changes for a standard OMPC algorithm without c_∞.

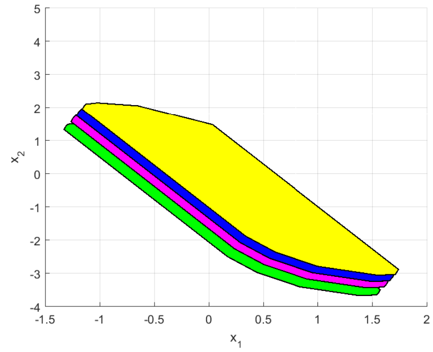

Fig. 1. Variation in feasible region of example (14) with $n_c = 2, r_{k+1} = 0$, no c_∞ and $u_{k-1} = 0.5, 0.2, 0, -0.2$.

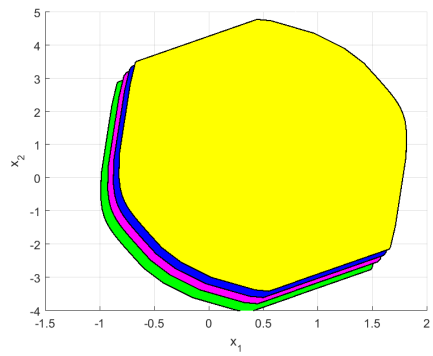

Fig. 2. Variation in feasible region of example (14) with $n_c = 1, r_{k+1} = 0$ and $c_\infty \neq 0$ and $u_{k-1} = 0.5, 0.2, 0, -0.2$.

- Figures 2 show how the 1-step set changes for examples (14,15) as u_{k-1} changes and including the d.o.f. c_∞.

Two conclusions are obvious: (i) First it is essential that u_{k-1} is included as a parametric state and this can have a significant impact on whether a given x_k is feasible or not.(ii) Secondly, in this case, adding a d.o.f. c_∞ as opposed to c_{k+1} as given significant enlargements in the feasible region (all figures (1,2) and (3,4) have the same number of d.o.f. but clearly the latter of each pair has larger volumes.).

B. With and without c_∞ but varying r_k

Again, the literature has tended to focus on feasible regions where the concern is focussed around the initial condition and regulation with an almost tacit assumption that the target is the origin. In practice, the target may change and this can have significant effects on the shape of the feasible region. In such a case, the traditional OMPC d.o.f., that is $\underset{\rightarrow k}{c}$ may, or may not, be effective.

This section uses example (14) and shows how the feasible region shape and volume changes substantially as the target changes and moreover emphasises that the standard d.o.f. in $\underset{\rightarrow k}{c}$ may have a limited impact in dealing with this.

- Figure 5 shows how the 2-step set changes as r_{k+1} changes for a standard OMPC algorithm without c_∞.

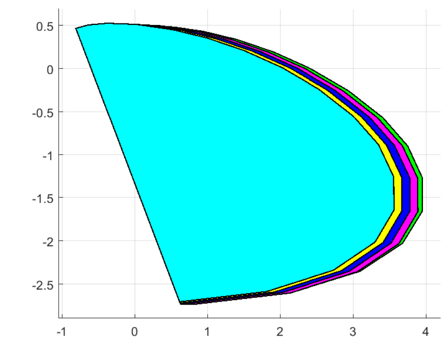

Fig. 3. Variation in feasible region of example (15) with $n_c = 1, r_{k+1} = 0$ no c_∞ and $u_{k-1} = 2, 1, 0, -1, -2$.

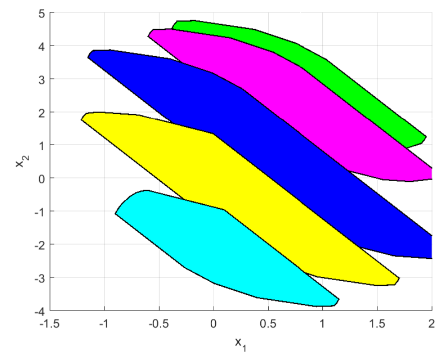

Fig. 5. Variation in feasible region of example (14) with $n_c = 2, u_{k-1} = 0$ and $r_{k+1} = -1, -0.5, 0, 0.5, 1, 1.2$.

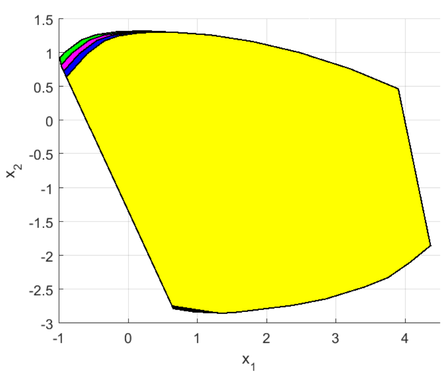

Fig. 4. Variation in feasible region of example (15) with $n_c = 1, r_{k+1} = 0$ and $c_\infty \neq 0$ and $u_{k-1} = 2, 1, 0, -1, -2$.

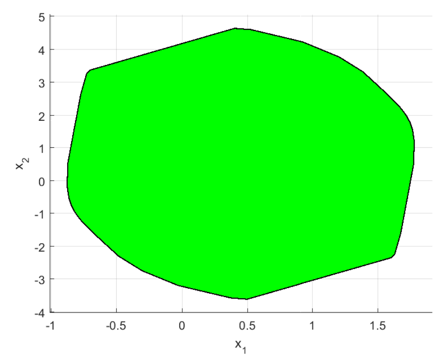

Fig. 6. Variation in feasible region of example (14) with $n_c = 1, u_{k-1} = 0$ and $c_\infty \neq 0$ and $r_{k+1} = -1, -0.5, 0, 0.5, 1, 1.2$.

- Figure 6 shows how the 1-step set changes as r_{k+1} changes but also including the d.o.f. c_∞.

It is notable here that the 2nd option, the algorithm which includes c_∞, has a feasible region which is totally unaffected by the choice of r_{k+1}. In retrospect this is to be expected, but of course it demonstrates the huge benefit of this option as opposed to the conventional OMPC algorithm whose feasible regions (figure 5) are much smaller by comparison, with the inevitable risk that frequent infeasible scenarios could arise.

Similar conclusions are derived by looking at example (15) - see figure 7, although it is interesting that for this example the feasible region volume is closely linked to the choice of target steady-state and some choices of target (e.g. $r = 1.2$) provide a feasible region that encompasses the smaller feasible regions for alternative choices of r. However, once again the most significant point is that if c_∞ is included as a d.o.f., the feasible region is unaffected by the choice of r.

V. Parametric solutions

While the previous section has shown potential feasibility benefits of exploiting the offset term in a closed-loop prediction paradigm, another interesting question is whether this d.o.f. can be used to simplify the complexity of a parametric solution. That is, if one can obtain a similar volume feasible region with far fewer d.o.f., is it possible that one may also require far fewer parametric regions.

In this case it is not obvious that any analysis or theorems will offer insight, but the authors consider it interesting to perform some case studies to investigate whether there were any encouraging patterns. The basic premise is to find the complexity of the associated parametric solution. Where c_∞ is included, the implied number of d.o.f. is one higher and of course the volumes of the feasible regions differ, but here the focus is solely on the parametric solution complexity.

The results are presented in tables I-II and use the same examples as in the previous section; it is accepted this is a very narrow snapshot and a far broader investigation is possible. A summary is that there is no obvious pattern, but of course one could argue that including c_∞ gives much larger feasible volumes in gernal for the same number of d.o.f., so for equivalent volumes of feasible region, it is likely that using c_∞ will result in far fewer parametric regions.

VI. Conclusions

This paper has made some investigations into the little studied area of feasibility in predictive control. A rather obvious but little explored insight is that the tool of a terminal set deployed in dual mode MPC to facilitate guarantees of stability can make feasibility worse as it introduces an

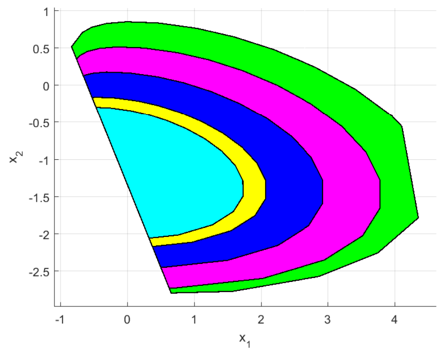

Fig. 7. Variation in feasible region of example (15) with $n_c = 1, r_{k+1} = 0$ and $c_\infty = 0$ and $r_{k+1} = -1, -0.5, 0, 0.5, 1, 1.2$.

Number d.o.f.	2	3	4	5	6
Without c_∞,$r_k = 0, u_{k-1} = 0$	32	58	79	105	142
With c_∞, $r_k = 0, u_{k-1} = 0$	32	58	79	105	189
Without c_∞, $r_k = 1, u_{k-1} = 0.5$	17	37	55	79	118
With c_∞, $r_k = 1, u_{k-1} = 0.5$	25	51	80	108	218

TABLE I

COMPARISON OF NUMBER OF REGIONS IN MPQP SOLUTION WITH D.O.F. OF JUST c_k AND WITH (c_k, c_∞) ON EXAMPLE (14).

artificial constraint on future predictions. Thus a means of relaxing this constraint without impacting on stability will improve feasibility.

It is known that the volume of the feasible regions is linked to the number of d.o.f., and indeed the choice of terminal control law, but little work has considered the dependence on parameters such as the target (desired steady-state) and the current input; indeed the visibility given to the importance of the current value of the input is a core contribution. Moreover, although reference governors are a common concept, these are rarely exploited in an integrated way into MPC algorithms. Here it is shown that the systematic inclusion of reference governor concepts, in essence the temporary move of the target, allows for potentially substantial increases in feasible volumes and thus caters for a number of important scenarios which otherwise could lead to infeasibility.

An a prioiri analysis of expected benefits cannot be performed in general, beyond the obvious scenario of set point changes, and will vary from case to case although the insights do apply equally to open stable and unstable processes. Moreover, future work should explore alternative reference governor paradigms as the current paper focuses on proposals which improve feasibility alone, but does not consider the impact on performance. Finally, there is a need to consider the repercussions for parametric solutions with a more comprehensive and wider ranging set of case studies.

Number d.o.f.	2	3	4	5	6
Without c_∞,$r_k = 0, u_{k-1} = 0$	40	86	138	193	255
With c_∞, $r_k = 0, u_{k-1} = 0$	40	86	138	193	355
Without c_∞, $r_k = 1, u_{k-1} = 2$	22	59	102	151	218
With c_∞, $r_k = 1, u_{k-1} = 2$	16	44	72	104	237

TABLE II

COMPARISON OF NUMBER OF REGIONS IN MPQP SOLUTION WITH D.O.F. OF JUST c_k AND WITH (c_k, c_∞) ON EXAMPLE (15).

Acknowledgements: Thanks to ETH for the use of the MPQP software toolbox http://control.ee.ethz.ch/mpt.

REFERENCES

[1] Aghaei, S., Sheikholeslam, F., Farina, M., and Scattolini, R. (2013). An mpc-based reference governor approach for offset-free control of constrained linear systems. *IJC*, 86, 1534–1539.

[2] Bemporad, A., Morari, M., Dua, V. and Pistikopoulos, E.N., (2002), The explicit linear quadratic regulator for constrained systems Automatica, 38 (1), pp. 320

[3] Bemporad, A. and Filippi,C. (2006), An algorithm for approximate multiparametric convex programming, Computational optimization and Applications, 35, pp-87-108.

[4] Clarke, D.W. and Mohtadi, C. (1989), Properties of generalized predictive control. *Automatica*, 25(6), 859–875.

[5] Dughman, S.S. and Rossiter, J.A. (2016), The feasibility of parametric approaches to predictive control when using far future feed forward information, submitted.

[6] Gilbert, E. and Tan, K. (1991), Linear systems with state and control constraints: The theory and application of maximal admissible sets. *IEEE Transactions Automatic Control*, 36, 1008–1020.

[7] Gilbert, E.G. and Kolmanovsky, I. (1995), Discrete time reference governors and the non-linear control of systems with state and control constraints, *Int. J. Robust and non-linear control*, 5, 487-504.

[8] Khan, B. and Rossiter, J.A. (2013), Alternative Parameterisation within predictive Control: a systematic selection, IJC, 86, 1397–1409. *Automatica*, 34(1), 65–73.

[9] Kouramas, K.I., Faisca, N.P., Panos, C. and Pistokopoulos, E.N. (2011), Explicit/multi-parametric model predictive control (MPC) of linear discrete-time systems by dynamic and multi-parametric programming, Automatica, 47, pp1638–1645.

[10] Kouvaritakis, B., Rossiter, J.A. and Schuurmans, J. (2000), Efficient robust predictive control, IEEE Trans. AC, 45, 8, pp1545-1549.

[11] Limón, D., Alvarado, I., Alamo, T., and Camacho, E.F. (2008). Mpc for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9), 2382–2387.

[12] Mayne, D.Q., Rawlings, J.B., Rao, C.C. and Scokaert, P.O. (2000), Constrained model predictive control: feasibility and optimality. *Automatica*, 36, 789–814.

[13] Pistikopoulos, E.N., Dua, V., Bozinis, N.A., Bemporad, A., Morari, M. (2002), On-line optimization via off-line parametric optimization tools Computers and Chemical Engineering, 26 (2), pp. 175185

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

[15] Rawlings, J.B., Bonn, D., Jrgensen, J., Venkat, A., and Jrgensen, S. (2008). Unreachable setpoints in model predictive control. *IEEE Trans. AC*, 53(9), 2209–2215.

[16] Rossiter, J.A., Gossner, J.R. and Kouvaritakis, B. (1996), Guaranteeing feasibility in constrained stable generalised predictive control, Proc. IEE Pt.D, 143, 5, pp463-469

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

[18] Rossiter, J.A., Kouvaritakis, B. and Cannon, M. (2005), An algorithm for reducing complexity in parametric predictive control, IJC, 78, 18, pp1511–1520

[19] Rossiter, J.A. (2006), A global approach to feasibility in linear MPC, ICC.

[20] Rossiter, J.A., Wang, L. and Valencia-Palomo,G. (2010), Efficient algorithms for trading off feasibility and performance in predictive control, IJC, 83,4,789.

[21] Shead, L., Muske, K., and Rossiter, J. (2010). Conditions for which mpc fails to converge to the correct target. *JPC*, 20, 1207–1219.

[22] Scokaert, P.O. and Rawlings, J.B. (1998), Constrained linear quadratic regulation. *IEEE Trans Auto. Cont*, 43(8), 1163–1169.