Comparisons and combinations of interpolation methods with conventional predictive control

Yihang Ding* and J.A Rossiter*
*Automatic Control and System Engineering, University of Sheffield, UK

Abstract—Many papers have demonstrated that interpolation methods can often enlarge feasible regions significantly while also reducing computational complexity in MPC algorithms (Bacic et al, 2003; Rossiter et al, 2004). However, the simplest interpolations give non-convex feasible regions and thus may have limited benefit in some state directions. This paper presents a new interpolation method which combines the advantages of simple interpolations with conventional predictive control methods to enable a better compromise balance between the volume and shape of the feasible region and the number of degrees of freedom required.

Keywords: Constraints, interpolation, feasibility, computational efficiency

I. INTRODUCTION

Model Predictive Control (MPC) (Camacho et al, 2005; Rossiter, 2003), is one of the most important advanced control techniques to have had a significant and widespread impact on industrial process control. Within this approach, a common objective is to guarantee asymptotic stability and recursive constraint satisfaction for a set of initial states that is as large as possible and with both a minimal control cost and computational load. Hence, some major issues with the implementation of MPC are: i) constraint handling usually requires an online optimizer which may imply significant computation; (ii) the feasible region within which the control law is well defined, may be small unless the algorithm uses large numbers of degrees of freedom (d.o.f.) and (iii) one can sometimes enlarge the feasible region by detuning, but this could be undesirable. From these points of view, there is typically a conflict existing between computational efficiency which depends on the number of d.o.f., the volume of the feasible region and the performance. Interpolation techniques (Rossiter et al, 2004; Bacic et al, 2003) provide one means of trading off between computation and feasibility and hence it is on these that the paper focuses.

This paper serves two purposes. First it gives a brief overview of some latest existing interpolation techniques (Bacic et al, 2003; Rossiter et al, 2004; Rossiter et al, 2005; Yihang Ding and Rossiter et al, 2006) developed to reduce the computational complexity by formulating classes of predictions with small numbers of d.o.f.. These algorithms can be summarised into two simple classes; those that use a specified number of d.o.f. (e.g. 1 or 2) and those where the number of d.o.f. is linked to the state dimension nx. Clearly, even though the latter class may give larger feasible regions, the use of nx d.o.f. is not computationally efficient for large dimensional systems. Hence the second contribution of this paper is to propose a novel interpolation method that combines the benefits of simpler interpolation methods with conventional predictive control e.g. (Scokaert et al, 1998) (OMPC).

The problem with conventional algorithms is that, being based on a single feedback law, each extra d.o.f. may have a limited impact on feasibility (Rossiter et al, 2004), although of course the benefit is a well posed optimisation and thus an assurance of good behaviour within the feasible region. Moreover, in so far as this is possible, the d.o.f. expand the feasible region in all directions, albeit slowly. Interpolation, by allowing different underlying control laws, can give relatively large feasibility improvements (Tan et al, 1992; Rossiter et al, 2004) with just a few d.o.f., but the expansion of the feasibility region, even if large overall, is much more directional.

This paper is the first to our knowledge that looks at forming a control law that combines both approaches systematically into an algorithm that nevertheless requires only a quadratic programming optimisation. It will be shown by example that the proposed approach does indeed form a middle ground between interpolation and OMPC. It is noted that this paper considers combinations with so called ‘ONEDOF’ algorithms (Rossiter et al, 2004); the concept can be extended to consider other interpolations but that is not covered here.

Section 2 will give some background to MPC, the modelling assumptions and then gives a detailed summary of several interpolation algorithms. Section 3 develops the proposed combination approach and Section 4 gives two illustrative examples. The paper finishes with conclusions and plans for future work.

II. BACKGROUND

This section introduces standard material from the existing literature on MPC, invariant sets and some basic interpolation schemes. The interpolations vary from those requiring at least nx d.o.f. (GIMPC, GIMPC2) to those requiring just one or two d.o.f. (ONEDOF, GIMPC2β). For completeness, the OMPC algorithm is also defined briefly.

A. Model and objective

This paper considers linear systems of the form

\[ x(k+1) = Ax(k) + Bu(k), \quad k = 0, \ldots, \infty \]  \hspace{1cm} (1)

and subject to constraints

\[ u(k) \in U \equiv \{ u : u \leq u \leq \pi \}, \quad k = 0, \ldots, \infty, \]  \hspace{1cm} (2a)

\[ x(k) \in X \equiv \{ x : x \leq x \leq \pi \}, \quad k = 0, \ldots, \infty. \]  \hspace{1cm} (2b)

Abstract—Many papers have demonstrated that interpolation methods can often enlarge feasible regions significantly while also reducing computational complexity in MPC algorithms (Bacic et al, 2003; Rossiter et al, 2004). However, the simplest interpolations give non-convex feasible regions and thus may have limited benefit in some state directions. This paper presents a new interpolation method which combines the advantages of simple interpolations with conventional predictive control methods to enable a better compromise balance between the volume and shape of the feasible region and the number of degrees of freedom required.

Keywords: Constraints, interpolation, feasibility, computational efficiency

I. INTRODUCTION

Model Predictive Control (MPC) (Camacho et al, 2005; Rossiter, 2003), is one of the most important advanced control techniques to have had a significant and widespread impact on industrial process control. Within this approach, a common objective is to guarantee asymptotic stability and recursive constraint satisfaction for a set of initial states that is as large as possible and with both a minimal control cost and computational load. Hence, some major issues with the implementation of MPC are: i) constraint handling usually requires an online optimizer which may imply significant computation; (ii) the feasible region within which the control law is well defined, may be small unless the algorithm uses large numbers of degrees of freedom (d.o.f.) and (iii) one can sometimes enlarge the feasible region by detuning, but this could be undesirable. From these points of view, there is typically a conflict existing between computational efficiency which depends on the number of d.o.f., the volume of the feasible region and the performance. Interpolation techniques (Rossiter et al, 2004; Bacic et al, 2003) provide one means of trading off between computation and feasibility and hence it is on these that the paper focuses.

This paper serves two purposes. First it gives a brief overview of some latest existing interpolation techniques (Bacic et al, 2003; Rossiter et al, 2004; Rossiter et al, 2005; Yihang Ding and Rossiter et al, 2006) developed to reduce the computational complexity by formulating classes of predictions with small numbers of d.o.f.. These algorithms can be summarised into two simple classes; those that use a specified number of d.o.f. (e.g. 1 or 2) and those where the number of d.o.f. is linked to the state dimension nx. Clearly, even though the latter class may give larger feasible regions, the use of nx d.o.f. is not computationally efficient for large dimensional systems. Hence the second contribution of this paper is to propose a novel interpolation method that combines the benefits of simpler interpolation methods with conventional predictive control e.g. (Scokaert et al, 1998) (OMPC).

The problem with conventional algorithms is that, being based on a single feedback law, each extra d.o.f. may have a limited impact on feasibility (Rossiter et al, 2004), although of course the benefit is a well posed optimisation and thus an assurance of good behaviour within the feasible region. Moreover, in so far as this is possible, the d.o.f. expand the feasible region in all directions, albeit slowly. Interpolation, by allowing different underlying control laws, can give relatively large feasibility improvements (Tan et al, 1992; Rossiter et al, 2004) with just a few d.o.f., but the expansion of the feasibility region, even if large overall, is much more directional.

This paper is the first to our knowledge that looks at forming a control law that combines both approaches systematically into an algorithm that nevertheless requires only a quadratic programming optimisation. It will be shown by example that the proposed approach does indeed form a middle ground between interpolation and OMPC. It is noted that this paper considers combinations with so called ‘ONEDOF’ algorithms (Rossiter et al, 2004); the concept can be extended to consider other interpolations but that is not covered here.

Section 2 will give some background to MPC, the modelling assumptions and then gives a detailed summary of several interpolation algorithms. Section 3 develops the proposed combination approach and Section 4 gives two illustrative examples. The paper finishes with conclusions and plans for future work.

II. BACKGROUND

This section introduces standard material from the existing literature on MPC, invariant sets and some basic interpolation schemes. The interpolations vary from those requiring at least nx d.o.f. (GIMPC, GIMPC2) to those requiring just one or two d.o.f. (ONEDOF, GIMPC2β). For completeness, the OMPC algorithm is also defined briefly.

A. Model and objective

This paper considers linear systems of the form

\[ x(k+1) = Ax(k) + Bu(k), \quad k = 0, \ldots, \infty \]  \hspace{1cm} (1)

and subject to constraints

\[ u(k) \in U \equiv \{ u : u \leq u \leq \pi \}, \quad k = 0, \ldots, \infty, \]  \hspace{1cm} (2a)

\[ x(k) \in X \equiv \{ x : x \leq x \leq \pi \}, \quad k = 0, \ldots, \infty. \]  \hspace{1cm} (2b)
\[ x(k) \in \mathbb{R}^{n_x} \text{ and } u(k) \in \mathbb{R}^{n_u} \text{ denote state and input vectors at discrete time } k \text{ with } n_x \text{ and } n_u \text{ respectively denoting the number of states and inputs of the system.} \]

More general linear state, input and mixed state/input constraints can also be considered without significantly complicating further sections. In this paper a new algorithm is proposed that stabilizes system (1) and guarantees satisfaction of constraints (2). The algorithm aims to minimize a performance index of the form:

\[ J = \sum_{k=0}^{\infty} (x(k)^T Q x(k) + u(k)^T R u(k)) \]  

with \( Q \in \mathbb{R}^{n_x \times n_x} \) and \( R \in \mathbb{R}^{n_u \times n_u} \) positive definite state and input cost weighting matrices.

**B. Invariant Sets - the nominal case**

It is assumed throughout this paper that any sets used are not only invariant (Blanchini, 1999), but in general are the maximal admissible sets (MAS, (Gilbert et al., 1991)) corresponding to any given feedback or interpolation. Some key definitions are thus given next.

**Definition 2.1 (Feasibility):** Given system (1), an asymptotically stabilizing feedback \( u(k) = -K x(k) \) and constraints (2), then a set \( S \subseteq \mathbb{R}^{n_x} \) is feasible if \( S \subseteq \{ x \in X, -K x \in U \} \).

**Definition 2.2 (Invariance):** Given system (1), a stabilizing feedback \( u(k) = -K x(k) \) and constraints (2), then a set \( S \subseteq \mathbb{R}^{n_x} \) is positive invariant iff \( x \in S \Rightarrow (A - BK)x \in S \).

**Definition 2.3 (MAS and predictions):** Under certain convergence conditions, the MAS for an LTI system (1) is given by \( S = \bigcap_{k=0}^{\infty} \{ x \in X, -K(A - BK)^k x \in U \} \) with \( n \) a finite number (thus \( S \) is polyhedral in general).

1) \( S_i \) is the MAS associated to feedback \( u = -K_i x \).

\[ S_i = \{ x : M_i x \leq d \}; \quad \lambda_s i \equiv \{ x : M_i x \leq \lambda d \}; \]  

Assume the origin is strictly inside \( S_i \) and hence normalise (5) so that \( d = [1, 1, \ldots, 1]^T \).

2) The closed-loop predictions for a given \( K \) are:

\[
\begin{align*}
    x(k) &= \Phi^k_i x(0) \\
u(k) &= -K_i \Phi^{-1} x(0)
\end{align*}
\]

\[ \Phi_i = A - BK_i \]  

**C. General Interpolation (GIMPC)**

This section summarises the algorithm of (Bacic et al., 2003; Rossiter et al., 2004). Given a system (1), constraints (2), a set of asymptotically stabilizing feedback controllers \( u(k) = -K_i x(k), i = 1, \ldots, n \) and corresponding invariant sets \( S_i \), consider the following decomposition:

\[ x(0) = \sum_{i=1}^{n} x_i, \text{ with } \sum_{i=1}^{n} \lambda_i = 1, \lambda_i \geq 0, x_i \in \lambda_i S_i \]

Given (7) (Bacic et al., 2003), define the GIMPC control law and associated feasible region \( \mathcal{S} \):

\[ u(k) = -\sum_{i=1}^{n} K_i x_i \]

\[ \mathcal{S} = \operatorname{Co}\{S_1, \ldots, S_n\} \]  

\[ \mathcal{S} \triangleq \operatorname{Co}\{S_1, \ldots, S_n\} \]

This law implies knowledge of the state decomposition in (7), the computation of which constitutes the main online computation. To do this, one needs to define and optimise the predicted performance. Thus:

1) define the input and state predictions as:

\[ u(k) = -\sum_{i=1}^{n} K_i \Phi_i x_i; \quad x(k) = \sum_{i=1}^{n} \Phi_i x_i \]

2) use Lyapunov theory to compute \( P \) (Bacic et al., 2003) such that the associated infinite-horizon cost is:

\[ J = \sum_{k=0}^{\infty} (x(k+1)^T Q x(k+1) + u(k)^T R u(k)) \]

(11)

\[ P = \Gamma_u R \Gamma_u^T + \Psi_i^T \Gamma_x \Psi_i + \Psi_i^T P \Psi_i \]

for \( i = 1, \ldots, n \) and

\[ \Psi_i = [(A_i - B_i K_i) \Psi_i + (A_i - B_i K_n) \Psi_i] \]

Algorithm 2.1 (GIMPC): Take a system (1), constraints (2), cost weighting matrices \( Q, R \), controllers \( K_i \) and invariant sets \( S_i \) and compute a proper \( P \). Then, at each time instant, given the state \( x(0) \), solve the following optimization:

\[ \min_{x_i, \lambda_i} \bar{x}^T P \bar{x}, \text{ subject to (7),} \]

Implement the input \( u = -\sum_{i=1}^{n} K_i x_i \).

Algorithm 2.1 guarantees recursive feasibility, constraint satisfaction and asymptotic stability and comprises algorithm 2.1 from (Rossiter et al., 2004) when the sets are defined as polyhedrals (Ellipsoids were used in (Bacic et al., 2003)).

**D. Extension of General interpolation (GIMPC2)**

The GIMPC could be highly conservative because the constraint handling implied by (7) is implicit rather than explicit. That is, this condition is sufficient to ensure constraint satisfaction, but not necessary. The conservatism arose from the condition \( x_i \in \lambda_i S_i \) and thus can be reduced by removing this condition and replacing it with explicit constraint handling (Rossiter et al., 2005). One can define the true feasible region from:

\[ S_{G2} = \{ x : \exists x_1, x_2, \text{ s.t. } M_1 x_1 + M_2 x_2 \leq d \} \]

(14)

which does not need the variable \( \lambda \) and associated constraints. However, mutually compatible \( M_1, M_2 \) are harder to define (this is offline), especially for the uncertain case.
Algorithm 2.2 (GIMPC2): At each time instant, given the current state $x$, solve the following optimisation problem:

$$\min_{x_i, i=1,\ldots,n} x^T P \dot{x}, \quad \text{subject to} \quad \begin{cases} M_1 x_1 + M_2 x_2 \leq d \\ x = x_1 + x_2 \end{cases}$$

and implement the input $u = -\sum_{i=1}^n K_i x_i$.

E. One degree of freedom interpolations (ONEDOF)

The disadvantage of GIMPC/GIMPC2 is that they require $n_\alpha$ d.o.f. so that they will be computationally efficient only for low dimensional systems. Simpler interpolation methods may deploy just one d.o.f. (Rossiter et al, 2004) and be based on just two possible control laws $K_1$, $K_2$.

ONEDOF interpolations obtain their simplicity by using co-linear interpolation. This means the state decomposition of (7) is restricted to:

$$x = x_1 + x_2; \quad x_1 = (1 - \alpha)x; \quad x_2 = \alpha x; \quad 0 \leq \alpha \leq 1$$

The associated control law is:

$$u = -[(1 - \alpha)K_1 + \alpha K_2] x.$$  \hspace{1cm} (17)

The predictions and thus constraint inequalities are constructed in the same manner as for GIMPC/GIMPC2, but using the $x_1$, $x_2$ defined in (16).

Algorithm 2.3 (ONEDOFa): \(\alpha\) is determined from:

$$\min \alpha \quad \text{s.t.} \quad [M_1(1 - \alpha) + M_2\alpha] x \leq d$$ \hspace{1cm} (18)

ONEDOFa can have a surprisingly large feasible region (Rossiter et al, 2004) but this region may be non-convex and give little benefit in some state directions. Where those directions are important, this algorithm could not be easily used.

F. Extension of GIMPC2 (GIMPC2\(\beta\))

The weakness of ONEDOF was the restriction to co-linear interpolation whereas the weakness of GIMPC2 was the need to allow a search over the entire state space. A compromise approach is to determine which search directions would give the most benefits for feasibility and this objective formed the basis of the GIMPC2\(\beta\) algorithm which uses two d.o.f., thus retaining computational efficiency even for large dimensional systems. The predictions and constraint inequalities are formed in the same way as for GIMPC, and only the underlying state decomposition is changed.

Define the decomposition (or search directions) as:

$$x = x_1 + x_2; \quad x_1 = (1 - \alpha)x + \beta \omega; \quad x_2 = \alpha x - \beta \omega;$$

The d.o.f. for the search are \(\alpha, \beta\) and the corresponding control law is

$$u = -[(1 - \alpha)K_1 + \alpha K_2] x - (K_2 - K_1) \beta \omega.$$ \hspace{1cm} (20)

One way of determining an 'optimum' search direction \(\omega\) is summarised in the following offline algorithm and this is following by the GIMPC2\(\beta\) control algorithm itself.

Algorithm 2.4 (The search direction \(\omega\)):

1) Define the maximum volume invariant ellipsoids $V_1$, $V_2$ (Kothare et al, 1996) for the given feedbacks $K_i$ and constraints (2) as:

$$V_i = \{x : x^T P_i x \leq 1\}; \quad i = 1,2$$ \hspace{1cm} (21)

2) Constraint satisfaction is ensured by (7) if $V \leq 1$, where

$$V = (x^T P_1 x)^{1/2} + (x^T P_2 x)^{1/2}$$ \hspace{1cm} (22)

3) In general, distance from constraints is maximised by minimising $V$. Thus suppose $x_1 = x + \omega; \quad x_2 = x - \omega$, then $V$ is minimised by:

$$\omega = 2P_1 x/(x^T P_2 x)^{1/2} - 2P_2 x/(x^T P_1 x)^{1/2}$$ \hspace{1cm} (23)

Algorithm 2.5 (GIMPC2\(\beta\)):

1) Define \(\omega\) from (23).
2) Update the constraint and cost from (24).
3) Minimise, w.r.t. \(\alpha, \beta\) the cost function subject to constraints:

$$\min_{\alpha, \beta} J = [\alpha \beta] S_{\alpha \beta} \left[\begin{array}{c} \alpha \\
\beta \end{array}\right] + [\alpha \beta] P_{\alpha \beta}$$

s.t. $[M_\alpha M_\beta] \left[\begin{array}{c} \alpha \\
\beta \end{array}\right] \leq d_{\alpha \beta}$ \hspace{1cm} (24)

4) Implement the control law as

$$u = -K_1((1 - \alpha)x + \beta \omega) - K_2(\alpha x - \beta \omega)$$ \hspace{1cm} (25)

G. O MPC or conventional MPC

OMPC is the algorithm of (Scokaert et al, 1998) modified as proposed in (Rossiter et al, 1998). The idea here is to start from the unconstrained optimal behaviour and then consider the degrees of freedom as perturbations $c(k)$ about the associated input trajectory. Hence the input predictions are defined, with a suitable optimum underlying state feedback $K$, as:

$$u(k + i) = -Kx(k + i) + c(k + i); \quad i = 0, \ldots, n_c - 1$$

$$u(k + n_c + i) = -Kx(k + n_c + i); \quad i \geq 0$$

and hence the d.o.f. for optimisation are summarised in $C = [c(k)^T, \ldots, c(k + n_c - 1)^T]^T$.

With an appropriate $K$ matching $J, W$ ($W > 0$), it is easy to show that optimisation of $J$ reduces to minimising $J = C^T W C$ subject to predictions satisfying constraints. The unconstrained optimum is $C = 0$, which will be used when feasible.

This algorithm can encompass the global optimum, if $n_c$ is large enough. However, the feasible region can be small, if $K$ is well tuned, for small values of $n_c$.

Remark 2.1: The predictions based on feedback $K_1$ satisfy constraints (2) if inequalities of the form:

$$M_1 x + N_1 C \leq d$$ \hspace{1cm} (27)

are satisfied, for $M_1$ defined the same as earlier and an appropriate $N_1$ (Rossiter, 2003).
III. GIMPC2C: A NEW INTERPOLATION ALGORITHM

The previous section has summarised several possible approaches to constraint handling.

- OMPC allows optimum constrained performance for a well tuned $K$, but may require large numbers of d.o.f. to give reasonable feasible regions. However, it does allow any choice of numbers of d.o.f., with consequent restrictions to feasibility.
- GIMPC2 (and GIMPC) are interpolations that can give large increases in feasibility by a simple mixing of two control laws. However, where the state dimension is large, this too would require large numbers of d.o.f.
- ONEDOF and GIMPC2β are restricted to just 1 and 2 d.o.f respectively but nevertheless can give large feasible regions. However, the increases can be direction dependent. Stability proofs/properties are also less straightforward to establish.

The intent here is to consider how one might combine the benefits of OMPC and ONEDOF. That is, use ONEDOF to give a large increase in feasibility and the flexibility in the OMPC d.o.f. to ensure there is some increase in all directions, where this might be required, without adding many extra d.o.f.. To the authors knowledge, no-one else has yet looked at how these two approaches can be combined into a single algorithm.

A. Proposal summary and prediction

Our proposal is to combine co-linear interpolation with the control law of (26). To do this, it is necessary to use the linearity property. Thus, consider different components of the prediction independently and then summate the results later.

First, consider the ONEDOF predictions in detail. The decomposition is given as:

$$ x(k) = x_1(k) + x_2(k) $$
$$ x_1 = (1 - \alpha)x $$
$$ x_2 = \alpha x $$

with associated predictions

$$ x(k + i) = \Phi x_i(k) + \Phi x_2(k) $$
$$ u(k + i) = -K x_1(k + i) - K x_2(k + i) $$

Substituting in from (28) gives:

$$ x(k + i) = [(1 - \alpha)\Phi_1 + \alpha\Phi_2]x(k) $$
$$ u(k + i) = -[(1 - \alpha)K_1 x_1(k + i) + \alpha K_2]x(k + i) $$

Second, consider the effect of perturbations $c(k)$ upon the predictions, assuming that after implementation these are fed back through feedback gain $K$. For simplicity, detail this component as $x_3(k)$, and therefore:

$$ x_3(k + 1) = Bc(k) $$
$$ x_3(k + 2) = \Phi_1 Bc(k) + Bc(k + 1) $$
$$ x_3(k + 3) = \Phi_1^2 Bc(k) + \Phi_1 Bc(k + 1) + Bc(k + 2) $$

It is obvious (Rossiter, 2003) that a more general form ($n > n_c$) is:

$$ x_3(k + n) = [\Phi_1^{n-1} B, \Phi_1^{n-2} B, \ldots, \Phi_1^{n-n_c} B]c $$
$$ u_3(k + n) = -[K\Phi_1^{n-1} B, K\Phi_1^{n-2} B, \ldots, K\Phi_1^{n-n_c} B]c $$

Combining the predictions (30,32), using linearity gives

$$ x(k + i) = [(1 - \alpha)\Phi_1 + \alpha\Phi_2]x(k) $$
$$ u(k + i) = -[(1 - \alpha)K_1 x_1(k + i) + \alpha K_2]x(k + i) $$

The following Lemma and Corollary are obvious and thus given without proof.

**Lemma 3.1:** The predictions are linear in the degrees of freedom $\alpha$ and $C$.

**Corollary 3.1:** A predictive control law making use of predictions (33) will give rise to a quadratic program where the optimisation variables are $\alpha, C$. The number of d.o.f. will be $1 + n_c + n_a$, where $n_a$ is the input dimension and $n_c$ is the block dimension of $C$.

B. The proposed GIMPC2C algorithm

Given that the constraints and performance index are the same as for GIMPC2, then substituting in from (18,33,27), the constraints take the form:

$$ [M_\alpha M_c] \begin{bmatrix} \alpha \\ C \end{bmatrix} \leq d_{ac} $$

$$ M_\alpha = [M_2 - M_1]x, \ M_c = N_1, \ d_{ac} = d - M_1x. $$

and the performance index reduces to:

$$ J = [\alpha C]S_{ac} \begin{bmatrix} \alpha \\ C \end{bmatrix} + [\alpha C]P_{ac} $$

for appropriate $S_{ac}, P_{ac}$ (details omitted).

**Remark 3.1:** In the unconstrained case, the optimum must be $\alpha = 0, C = 0$ and hence one can deduce that $P_{ac} = 0$.

**Algorithm 3.1 (GIMPC2C):**

1) Update the constraint and performance index from (34,35).

2) Minimise $J$, w.r.t. $\alpha, C$ and the subject to constraints:

$$ \min_{\alpha,C} J = [\alpha C]S_{ac} \begin{bmatrix} \alpha \\ C \end{bmatrix} + [\alpha C]P_{ac} $$

s.t. $[M_\alpha M_c] \begin{bmatrix} \alpha \\ C \end{bmatrix} \leq d_{ac} $(36)

3) Implement the control law as

$$ u(k) = -[(1 - \alpha)K_1 + \alpha K_2]x(k) + c(k) $$

where $c(k)$ is the first component of $C$.

**Remark 3.2:** As with ONEDOF, a generic proof of stability is easiest if one assumes that one can always ride on the tail (Mendez et al, 2000), so a simple proof can be established with the following rule. If the new optimum does not give a lower cost than the tail from the previous sample, use the tail which by definition must give a reduction in cost. Thus the cost is always monotonically decreasing (in the nominal case).
IV. EXAMPLE

In this section two examples with two states and three states are used to prove the efficiency of the proposed GIMPC2C. Particularly illustrations are given of the associated feasible region, closed-loop performance and computational load in comparison with the pre-existing interpolation methods.

A. Example 1

The model and constraints are given by:

\[
A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix} \quad (38) \\
C_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad D_1 = 0 \quad (39)
\]

\[
\pi = 1, \quad \mu = -1 \quad (40) \\
\sigma = [2, 2]^T, \quad x = [-2, -2]^T \quad (41)
\]

The LQR-optimal controller is derived with \( Q = \text{diag}(1, 0), R = 0.1 \).

**Feasible Regions:** Figures 1, 2 give the underlying MAS \( S_1, S_2 \), the feasible regions for GIMPC2, GIMPC2\( \beta \), OMPC and GIMPC2\( C \) of 2-state example. It is clear that in this case GIMPC2\( C \) improves upon the feasibility of GIMPC, and has quite similar feasibility with GIMPC2\( \beta \), but less feasibility than GIMPC2. The key point however is that the proposed algorithm has improved significantly on both \( S_1, S_2 \) and OMPC with \( n_c = 2 \). Figures 3, 4 and 5 give the same comparison of these algorithms of 3-state example. The conclusions are quite similar with the ones in 2-state example.

**Control Performance:**

<table>
<thead>
<tr>
<th>GIMPC2 C</th>
<th>GIMPC2 ( \beta )</th>
<th>GIMPC2</th>
<th>OMPC ( (n_c = 2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0174</td>
<td>1.02517</td>
<td>1.02238</td>
<td>1</td>
</tr>
</tbody>
</table>

**TABLE 1**

**Normalized average costs**

B. Example 2

This model and constraints are given by:

\[
A_2 = \begin{bmatrix} 0.98 & 0 & 0.019 \\ 0.075 & 0.607 & 0.001 \\ 0 & 0 & 0.607 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0.005 \\ -0.021 \\ 0.39 \end{bmatrix} \quad (42) \\
C_2 = \begin{bmatrix} 1.69; 13.22; 0 \end{bmatrix}, \quad D_2 = 0 \quad (43)
\]

\[
\pi = 1, \quad \mu = -1 \quad (44) \\
\sigma = [2, 2]^T, \quad x = [-2, -2]^T \quad (45)
\]

The LQR-optimal controller is derived with \( Q = \text{diag}(1, 0), R = 0.1 \).

**Computational Load:** The table 2 is the comparison of the numbers of d.o.f utilised by each algorithm

V. CONCLUSION AND FUTURE WORK

This paper makes a novel contribution in the field of interpolation based MPC system. The observation is
made that existing MPC techniques are polarized into interpolation methods using full state decomposition or just 1,2 directions and standard MPC laws using perturbations about a fixed control law. This paper has proposed a systematic methodology for combining interpolation methods with the control perturbation approach, thus increasing flexibility of the design to improve feasibility, while retaining the fundamental motivation of low computational burdens.

Nevertheless, the proposal in this paper constitutes preliminary results and needs more rigorous testing and development. In particular, the intention is to consider higher order examples and extensions to cater for model uncertainty.

VI. ACKNOWLEDGMENTS
This principal investigator is funded by ORS, the University of Sheffield and the ACSE department.

REFERENCES
Camacho, E. and C. Bordons, Model predictive Control, Springer Verlag