ENLARGING FEASIBLE REGIONS FOR INTERPOLATION METHODS

Yihang Ding and J.A Rossiter *

* University of Sheffield
Department of Automatic Control and Systems Engineering

Abstract: Interpolation approaches can be used to reduce computational complexity in MPC algorithms (Bacic et al., 2003; Rossiter et al., 2004), but this can be at a cost to the feasible regions. Nevertheless recent work (Rossiter et al., 2005) demonstrated that interpolation can enlarge feasible regions far more widely than originally thought. This however required the use of several degrees of freedom. This paper presents a new decomposition method which combines the insights of several existing interpolations to achieve a balance between the volume of the feasible region and the number of degrees of freedom required.

Keywords: constraints, interpolation, feasibility, computational efficiency

1. 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. A common objective in the MPC community is guaranteeing 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. Interpolation techniques (Rossiter et al., 2004; Bacic et al., 2003) provide a favorable trade off between these different aspects. For instance, some major obstacles to 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.

This paper serves two purposes. First it gives a brief overview of some existing interpolation techniques (Bacic et al., 2003; Rossiter et al., 2004; Rossiter et al., 2005) 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 one d.o.f. and those which require a number of d.o.f. roughly equal to the state dimension $n_x$. Clearly, even though the latter class may give large feasible regions, the use of $n_x$ d.o.f. is not computationally efficient for large dimensional systems. Hence the second contribution of this paper is to propose a new interpolation method that makes use of the best facets of the existing ones. For this paper, this interpolation will be restricted to just two d.o.f. and thus could be considered highly efficient, when its feasible region is large.

Section 2 will give some background to MPC, the modelling assumptions and previous work.
Section 3 then gives insight on the definition of invariant sets to formulate an algorithm using just two interpolation directions. Section 4 gives a simulation example and the paper finishes with conclusions and plans for future work.

2. BACKGROUND

This section introduces standard material from the existing literature on MPC, invariant sets and some basic interpolation schemes.

2.1 Model and objective

This paper considers linear systems of the form

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

and subject to constraints

\[ u(k) \in U \equiv \{ u : \underline{u} \leq u \leq \bar{u} \}, \quad k = 0, \ldots, \infty, \]  

\[ x(k) \in X \equiv \{ x : \underline{x} \leq x \leq \bar{x} \}, \quad k = 0, \ldots, \infty. \]  

As denoted state and input vectors at discrete time \( k \) with \( n_x \) and \( n_u \) 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 minimise

\[ J = \sum_{k=0}^{\infty} (x(k)^TQx(k) + u(k)^TRu(k)) \]  

as a cost objective 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.

2.2 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 \( u(k) = -Kx(k) \) and constraints (2), then a set \( S \subset \mathbb{R}^{n_x} \) is positive invariant iff \( S \subset \{ x | x \in X, -Kx \in U \} \).

**Definition 2.2. (Invariance).** Given system (1), a stabilizing feedback \( u(k) = -Kx(k) \) and constraints (2), then a set \( S \subset \mathbb{R}^{n_x} \) is positive invariant iff

\[ x \in S \Rightarrow (A - BK)x \in S. \]  

This paper makes use of the MAS which is the largest possible feasible and invariant set. Under certain convergence conditions, the MAS for an LTI system ((1)) is given as \( S = \bigcap_{k=0}^{n} \{ x(A - BK)^{k}x \in X, -K(A - BK)^{k}x \in U \} \) with \( n \) a finite number (thus \( S \) is polyhedral in general).

**Remark 1.** In the following discussions, we will assume that the matrices \( M_i \) defining \( S_i \) are defined in a mutually consistent fashion. Hence the jth row of each, for all \( j \), must refer to the same constraint. Consequently, the sets \( S_i \) may be non-minimal as redundant constraints are only removed if they are redundant for every \( S_i \).

**Definition 2.3. (MAS and predictions).**

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

\[ S_i = \{ x : M_i x \leq d \} \]  

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

(2) Introduce the shorthand notation:

\[ \lambda S_i \equiv \{ x : M_i x \leq \lambda d \} \]

(3) The closed-loop predictions for a given \( K_i \) are:

\[ x(k) = \Phi_i^k x(0); \quad u(k) = -K_i \Phi_i^{k+1} x(0); \quad \Phi_i = A - BK_i \]  

2.3 General Interpolation (GIMPC)

This section describes 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:

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

Furthermore, given (8) holds (Bacic et al, 2003), define the following control law and associated feasible region \( \mathcal{S} \):

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

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

More generally, it is necessary to determine, online, the decomposition implied in (8). To do this,
one needs to compute and optimise the predicted performance. Thus:

1. define the input and state predictions as:
   \[ u(k) = -\sum_{i=1}^{n} K_i \Phi_i^k x_i; \quad x(k) = \sum_{i=1}^{n} \Phi_i^k x_i. \]
   (11)

2. use Lyapunov theory to compute \( P \) and infinite-horizon cost (3) as [Appendix 1]:
   \[
   J = \bar{x}^T P\bar{x} = \sum_{k=0}^{\infty} x(k+1)^T Q x(k+1) + u(k)^T R u(k)
   \]
   where \( \bar{x} = [x_1^T \ldots x_n^T]^T \).
   (12)

**Algorithm 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 optimisation:

\[
\min_{x_i, \lambda} \bar{x}^T P \bar{x}, \quad \text{subject to } (8),
\]
where

\[
P = \Gamma_u^T R \Gamma_u + \Psi^T Q \Psi + \Psi^T P \Psi,
\]
for \( i = 1, \ldots, n \)

\[
\Psi_i = [(A_i - B_i K_1)^T \ldots (A_i - B_i K_n)^T]^T \\
\Gamma_x = [I, \ldots, I], \quad \Gamma_u = [K_1, \ldots, K_n]
\]
and implement the input \( u = -\sum_{i=1}^{n} K_i x_i \).

Algorithm 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)).

### 2.4 Extension of General interpolation (GIMPC2)

The GIMPC could be highly conservative because the constraint handling implied by (8) 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). Specifically, consider the more detailed definition of feasible region \( \bar{S} \):

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

The conservatism arises in the introduction of the \( \lambda \) term and associated constraints, which are unnecessary. An alternative feasible region, removing the \( \lambda \) constraint, is given as:

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

**Remark 2.** It is obvious that the feasible region \( \bar{S} \subseteq S_{G2} \). Hence \( S_{G2} \) gives a larger feasible region. Moreover, there are fewer d.o.f. as \( \lambda \) is no longer required. However, the set definitions required for \( \bar{S} \) may require far fewer inequalities than for \( S_{G2} \).

The associated algorithm, denoted GIMPC2, is given next.

**Algorithm 2.** (GIMPC2). At each time instant, given the current state \( x \), solve the following optimisation problem:

\[
\min_{x_i} \bar{x}^T P \bar{x}, \quad \text{subject to } \begin{cases} M_1 x_1 + M_2 x_2 \leq d \\ x = x_1 + x_2 \end{cases}
\]

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

**Remark 3.** Both GIMPC and GIMPC2 have guarantees of stability and recursive feasibility.

### 2.5 One degree of freedom interpolations and insights

The disadvantage of GIMPC/GIMPC2 is that they require \( n_x \) d.o.f. so that they will be computationally efficient only for low dimensional systems. An earlier alternative interpolation deploys just one d.o.f. (Rossiter et al, 2004) and uses just two possible control laws \( K_1, K_2 \). Here we give a brief summary of this approach, denoted ONEDOF for brevity.

ONEDOF interpolations deploy co-linear interpolation which means the state decomposition of (8) is restricted to those for which:

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

The control law can thus be simplified to:

\[
u = -[(1-\alpha)K_1 + \alpha K_2]x.
\]

Two algorithms have been proposed; these are analogous to GIMPC and GIMPC2 in that one has an implied constraint \( M_i x_i \leq \lambda_i \forall i, \sum_i \lambda_i = 1 \) and the other does not.

**Algorithm 3.** (ONEDOFa). \( \alpha \) is determined from:

\[
\min_{\alpha} \alpha \text{ s.t. } [M_1 (1-\alpha) + M_2 \alpha] x \leq d
\]

**Algorithm 4.** (ONEDOFb). \( \alpha \) is determined from an equivalent condition to (8):
3.1 Proposal summary

Our proposal is to allow two search directions in the decomposition of $x_t$, and thus make use of two d.o.f. This paper makes some introductory explorations to the potential of the specific choice for these two directions. So, specifically we define the search directions as the actual state $x$ and a new direction (to be defined) $\omega$, thus:

$$
\begin{align*}
   x &= x_1 + x_2 \\
   x_1 &= (1-\alpha)x + \beta \omega; \\
   x_2 &= \alpha x - \beta \omega;
\end{align*}
$$

(22)

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

$$
u = - [(1-\alpha)K_1 + \alpha K_2] x - (K_2 - K_1) \beta \omega.
$$

(23)

3.2 Selecting the best search direction

It is self evident that $x$ itself is a logical direction along which to interpolate. The choice of a second direction is less obvious. Naturally the best choice can be computed a posteriori from GIMPC2, but we must assume that such knowledge is not available and hence use insight and other tools to propose what we expect, but cannot guarantee a priori, will yield the most benefits.

The most obvious choice is a direction $\omega$ that is perpendicular to $x$. However, this is not unique for $n_x \geq 3$ and thus is not particular helpful in general. To summarise therefore:

**Problem:** Which direction $\omega$ is most likely to move the implied state trajectories for decomposition $x = x_1 + x_2$ (from (22)) further away from constraints.

**Tools available (insight):** For any given $x$, we know which direction moves one most rapidly away from constraints, that is move perpendicular to the facet of the invariant set at that value of $x$. We can measure how quickly one moves away from constraints using a simple norm computation.

For this paper, we will show how the above two statements and ellipsoidal invariant sets can be used to find a good direction, and moreover, one that can be defined analytically!

**Algorithm 5.** (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:

$$V_i = \{x : x^T P_i x \leq 1\}; \ i = 1, 2 \quad (24)$$

(2) Constraint satisfaction is ensured by (8) if $V \leq 1$, where

$$V = (x_i^T P_1 x_1)^{1/2} + (x_2^T P_2 x_2)^{1/2} \quad (25)$$

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

$$\omega = 2P_2 x/(x^T P_2 x)^{1/2} - 2P_1 x/(x^T P_1 x)^{1/2} \quad (26)$$

It should be noted that the optimisation makes use of inconsistent $x_1, x_2$ because $\alpha$ is no known either. The assumption is made that minimising $V_1$, $V_2$ starting from the same point will essentially give the best solution for $\alpha = 0.5$ and this should be close to what is wanted for other values of $\alpha$. Consideration to the benefits, or not, of more complex procedures is ongoing.

3.3 The proposed GIMPC$\beta$ algorithm

Once a suitable search direction has been determined, the interpolation algorithm is self evident. First note that the constraints and performance index are the same as for GIMPC2 and thus substituting in from (22), the constraints will be:

$$[M_\alpha \ M_\beta] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \leq d_{\alpha \beta} \quad (27)$$
Remark 4. (12) reduces to

\[ M = [M_2 - M_1]x, \quad M_\beta = [M_1 - M_2]\omega, \quad d_{\alpha\beta} = d - M_1x. \]

In a similar way, the performance index of (12) reduces to

\[ J = [\alpha \beta]S_{\alpha\beta} \left[ \begin{array}{c} \alpha \\ \beta \end{array} \right] + [\alpha \beta]P_{\alpha\beta} \quad (28) \]

for appropriate \( S_{\alpha\beta}, P_{\alpha\beta} \).

Algorithm 6. (GIMPC2\( \beta \)).

1. Define \( \omega \) from (26).
2. Update the constraint and cost from (27), (28).
3. Minimise, w.r.t. \( \alpha, \beta \) the cost function subject to constraints:

\[
\min_{\alpha,\beta} J = [\alpha \beta]S_{\alpha\beta} \left[ \begin{array}{c} \alpha \\ \beta \end{array} \right] + [\alpha \beta]P_{\alpha\beta} \\
\text{s.t.} \quad [M_\alpha M_\beta] \left[ \begin{array}{c} \alpha \\ \beta \end{array} \right] \leq d_{\alpha\beta} 
\]

(29)

4. Implement the control law as

\[ u = -K_1((1-\alpha)x + \beta\omega) - K_2(\alpha x - \beta\omega) \quad (30) \]

Remark 4. As with ONEDOF, a generic proof of recursive feasibility and convergence remains ongoing work. However, it is accepted 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).

4. EXAMPLE

In this section two examples with two and three states are used to prove the efficiency of the proposed GIMPC2(\( \beta \)). Particularly illustrations are given of the associated feasible region, closed-loop performance and computational load in comparison with the pre-existing interpolation methods.

4.1 Example 1

The model and constraints are given by:

\[
A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix} \quad (31) \\
C_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad D_1 = 0 \quad (32)
\]

\[
\tau = 1, \quad \alpha = -1 \quad (33) \\
\tau = [2, 2]^T, \quad \beta = [-2, -2]^T \quad (34)
\]

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

1. Feasible Regions: Figure 1(a) gives the underlying MAS \( S_1, S_2 \), the feasible region for GIMPC2 \( S_{G2} \) and the feasible region \( S_{G2b} \) for

![Diagram](image)

Fig. 1. Feasible region comparison of OMPC, GIMPC2 and GIMPC2 \( \beta \)

GIMPC2\( \beta \) (the feasible region for OMPC is equal to \( S_2 \) in this case). It is clear that in this case GIMPC2\( \beta \), has better feasibility than GIMPC and but clearly the restriction to the search direction has given less feasibility than GIMPC2. The key point however is that the proposed algorithm has improved on ONEDOF, whose feasible region is given by the union of \( S_1, S_2 \).

For completeness, figure 1(b) contains a comparison with a conventional MPC algorithm (Scokaert et al., 1998) deploying \( n_c \) d.o.f. and with a terminal region of \( S_1 \). It is clear that the gain in feasibility is similar to the use of \( n_c = [15, 20] \). Although in practice one might argue that the terminal control law is too highly tuned and a lower \( n_c \) would be required with an alternative \( K \), this is in fact simply restating one of the arguments behind interpolation.

2. Control Performance: In order to compare all three algorithms, we consider points inside the feasible regions of all the algorithms, but as close to the boundary as possible. Simulations are performed from a number initial conditions and the associated runtime costs are added normalised and averaged. The results are given in table 1. Perhaps surprisingly, in this case GIMPC2\( \beta \) has outperformed GIMPC2. However, of more significance is that the suboptimality in comparison with an ‘optimal’ control is quite small.

<table>
<thead>
<tr>
<th>( \text{GIMPC2} )</th>
<th>( \text{GIMPC2} )</th>
<th>( \text{OMPC} (n_c = 20) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0258</td>
<td>1.0320</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1. Normalized average costs

4.2 Example 2

This model and constraints are given by:
Fig. 2. Feasible region comparison of OMPC, GIMPC2 and GIMPC2b

The LQR-optimal controller is derived with $Q = \text{diag}(1, 0), R = 0.1$. (3)

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

<table>
<thead>
<tr>
<th>GIMPC2β</th>
<th>GIMPC2</th>
<th>GIMPC</th>
<th>OMPC ($n_c = 3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2. Numbers of degrees of freedom

5. CONCLUSION AND FUTURE WORK

This paper makes a novel contribution in the field of interpolation based MPC system. The observation is made that existing techniques are polarised into those using either one or $n_x$ d.o.f., probably because it is not clear in general how to set up an interpolation with an arbitrary number of d.o.f.. This paper has proposed a systematic methodology for a choosing two search directions, thus increasing feasibility as compared to ONEDOF algorithms, 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 to discuss much higher order examples is in the future.

It should be noted that although this paper is presented for the nominal case, the results will carry over to uncertain systems in a manner similar to that described in (Rossiter et al, 2005). Again, rigorous testing of this constitutes ongoing work.

6. ACKNOWLEDGMENTS

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

REFERENCES


Blanchini, F., Set Invariance in Control, Automatica, 1999, 35, 1747-1767

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


Kothare, M. V. and Balakrishnan, V. and Morari, M., Robust Constrained Model Predictive Control using Linear Matrix Inequalities, Automatica, 32, 1361-1379, 1996


Rossiter, J.A. and B. Pluymers, The potential of interpolation for simplifying predictive control and application to LPV systems, Workshop on NMPC, Stuttgart, 2005

Scokaert, P.O.M. and Rawlings, J.B., Constrained linear quadratic regulation, IEEE Transactions on Automatic Control, 43, 8, 1163-1168