
Proceedings Paper:
Valencia-Palomo, G., Rossiter, J.A. orcid.org/0000-0002-1336-0633, Jones, C.N. et al. (2 more authors) (2011) Alternative parameterisations for predictive control: how and why? In: Proceedings of the 2011 American Control Conference. 2011 American Control Conference, 29 Jun - 01 Jul 2011, San Francisco, CA, USA. IEEE , pp. 5175-5180. ISBN 9781457700804

https://doi.org/10.1109/ACC.2011.5990639

Alternative parameterisations for predictive control: how and why?

G. Valencia-Palomo, J.A. Rossiter, C.N. Jones, R. Gondhalekar and B. Khan.

Abstract— **This paper looks at the efficiency of the parameterisation of the degrees of freedom within an optimal predictive control algorithm. It is shown that the conventional approach of directly determining each individual future control move is not efficient in general, and can give poor feasibility when the number of degrees of freedom are limited. Two systematic alternatives are explored and both shown to be far more efficient in general.**

Keywords: MPC, Laguerre functions, feasibility volumes.

I. Introduction

Predictive control (MPC) [7], [10], [2] is popular because it handles multivariable processes with constraints in a systematic fashion, but to achieve this the online implementation may require a substantial optimisation (usually a QP). Here, there is a well understood set of potentially conflicting objectives, e.g. between the desire for good performance and large feasible regions with the equally important desire to keep the number of degrees of freedom (d.o.f.) small.

Recent works [14], [12] have shown that Laguerre polynomials are an effective alternative to the standard basis set for parameterising d.o.f. in the prediction set deployed by MPC. Specifically it was shown that in many cases changing the parameterisation allowed substantial improvements in feasibility with little or no detriment to performance. Nevertheless, one key question was still left unanswered: is there a systematic way of choosing the best 'Laguerre polynomial' or indeed is there an alternative to Laguerre which is better still? It is these questions which are tackled in this paper.

This paper will take the premise that the terminal control law should be well tuned and therefore this parameter is not available for influencing the size of feasible regions. Instead, the question is asked: how else can the designer increase the feasible region? The feasible region is sometimes called a reachable set or there is also the terminology of n-step sets. First define the maximal admissible set (MAS) [3] as the set of initial states from which the unconstrained *optimal* control law satisfies constraints indefinitely. A one step set is then the set of initial states from which there exists a feasible (i.e. satisfies constraints) control move which moves the state inside the MAS within one step. Similarly, a two step set can reach the MAS in at most two steps, again while satisfying constraints. The n-step sets assume some form of non-linear control strategy may be required due to the need for constraint satisfaction and these are often called maximal controlled admissible sets (MCAS).

Historically, most MPC algorithms used the individual predicted control values at future samples as the d.o.f. within the output and state predictions. However, while this seems to make sense, it is in fact very restrictive. For example, to grow the feasible region to a 10-step set would require 10 d.o.f. for each input, whereas in fact it may be possible to parameterise the required input trajectories far more efficiently. However, very little work has considered this particular issue. Two lines of enquiry have been interpolation methods and parametric programming [11], [1], but these both imply a significant change in either algorithm design or implementation and are not pursued here. Instead, this paper focuses on using a conventional MPC set up such as used in [13] and asks the simple question of can we parameterise the d.o.f. in the future input differently and if so, how and why?

One significant contribution here has been the concept of Triple mode control [5]. Here, the region immediately outside the MAS is controlled with a fixed linear time varying control law. This law can be determined using robust techniques first popularised in [6], thus one finds the best compromise between feasibility and performance. However, a critical weakness is that the choice of this law is based on ellipsoidal regions and these could be very suboptimal or a poor measure of feasibility in the polyhedral space. Nevertheless, it demonstrates a key concept, that the n-step set for large n can be reached with a parameterisation of future inputs that uses less than n d.o.f. per input. Moreover, it shows that this parameterisation can be linked to some form of underlying model.

One observation made in this paper is that MPC algorithms using Laguerre polynomials have greater overlap with Triple mode approaches than orginally realised - this overlap is demonstrated later in this paper where the flexibility in the assumed dynamic is discussed. Laguerre polynomials are a simple way of assuming dynamics with a given time constant, but also give an othornormal mapping of the available space and lead to well conditioned numerical problems.

The key question left to be resolved is, what is the best dynamic to assume for the predicted inputs, that is the d.o.f., to allow for large feasible regions? A secondary issue relates to this, what is the best initial condition for that dynamic and is this state dependent? This paper is organised as follows: after presenting the background in Section 2, Section 3 will show how one can incorporate any fixed dynamic into the predictions, and moreover this still gives recursive stability and feasibility, it will also show that the initial condition used for that dynamic can be arbitrary and thus is not an effective d.o.f.; Section 4 will then propose an alternative

G. Valencia-Palomo is with the Instituto Tecnológico de Hermosillo, México gvalencia@ith.mx. J.A. Rossiter and B. Khan are with the Department of Automatic Control and Systems Eng., University of Sheffield, UK. j.a.rossiter@shef.ac.uk, b.khan@shef.ac.uk. C.N. Jones is with the Automatic Control Laboratory, ETH, Switzerland. cjones@ee.ethz.ch. R. Gondhalekar is with the Frontier Research Base for Global Young Researcher and Dept. of Mechanical Engineering, Osaka University, Japan. ravi.gondhalekar@wakate.frc.eng.osaka-u.ac.jp.

mechanism for exploiting the desired input prediction space in an efficient manner but which is not necessarily linked to a fixed dynamic; Section 5 will present numerical examples. The paper finishes with the conclusions in Section 6.

II. Background

A. Modelling and Optimal MPC

Assume a standard state-space model of the form:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k; \quad \mathbf{y}_k = \mathbf{C}\mathbf{x}_k; \tag{1}$$

with $\mathbf{x}_k \in \mathbb{R}^{n_x}$, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ and $\mathbf{u}_k \in \mathbb{R}^{n_u}$ which are the state vector, the measured output and the plant input respectively. Associated to the model are polytopic constraints, e.g.

$$\underline{\mathbf{u}} \le \mathbf{u}_k \le \overline{\mathbf{u}}; \quad \underline{\Delta\mathbf{u}} \le \Delta\mathbf{u}_k \le \overline{\Delta\mathbf{u}}; \quad \underline{\mathbf{y}} \le \mathbf{y}_k \le \overline{\mathbf{y}}. \tag{2}$$

The performance index to be minimised (w.r.t. $\mathbf{u}_k, \mathbf{u}_{k+1}, \ldots$) is

$$\begin{array}{ll} J & = \sum_{i=0}^{\infty} (\mathbf{x}_{k+i+1})^T \mathbf{Q} (\mathbf{x}_{k+i+1}) + (\mathbf{u}_{k+i})^T \mathbf{R} (\mathbf{u}_{k+i}) \\ & s.t. \quad \begin{cases} (1), (2) \quad \forall k \ge 0, \\ \mathbf{u}_k = -\mathbf{K}\mathbf{x}_k \quad \forall k \ge n_c \end{cases} , \end{array} \tag{3}$$

with $\mathbf{Q}$, $\mathbf{R}$ positive definite state and input cost weighting matrices. $\mathbf{K}$ is the optimal feedback gain minimising J in the absence of constraints (2). Practical limitations imply that only a finite number, that is n_c, of free control moves can be used [13]. For these cases, (3) is implemented [8] by imposing that the state $\mathbf{x}_{n_c}$ must be contained in a polytopic control invariant set (that is the MAS): $\mathcal{X}_{MAS} = \{\mathbf{x}_0 \in \mathbb{R}^{n_x} \mid \underline{\mathbf{x}} \le \mathbf{x}_k \le \overline{\mathbf{x}}, \underline{\mathbf{u}} \le -\mathbf{K}\mathbf{x}_k \le \overline{\mathbf{u}}, \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \forall k \ge 0\}$. For simplicity of notation, the MAS can also be described in the form $\mathcal{X}_{MAS} = \{\mathbf{x}_k \in \mathbb{R}^{n_x} \mid \mathbf{M}\mathbf{x}_k \le \mathbf{d}\}$ for appropriate $\mathbf{M}, \mathbf{d}$.

For convenience, the degrees of freedom can be reformulated in terms of a new variable $\mathbf{c}_k$

$$\mathbf{u}_{k+i} = \begin{cases} -\mathbf{K}(\mathbf{x}_{k+i}) + \mathbf{c}_{k+i}; & i = 0, ..., n_c - 1; \\ -\mathbf{K}(\mathbf{x}_{k+n_c+i}); & i \ge 0; \end{cases} \tag{4}$$

and hence the equivalent optimisation to (3) is

$$\min_{\underrightarrow{\mathbf{c}}_k} \quad \underrightarrow{\mathbf{c}}_k^T \mathbf{S} \underrightarrow{\mathbf{c}}_k \quad s.t. \quad \mathbf{M}\mathbf{x}_k + \mathbf{N}\underrightarrow{\mathbf{c}}_k \le \mathbf{d}; \tag{5}$$

with $\underrightarrow{\mathbf{c}}_k = [\mathbf{c}_k^T, \ldots, \mathbf{c}_{k+n_c-1}^T]^T$. Details of how to compute positive definite matrix $\mathbf{S}$, matrices $\mathbf{N}$, $\mathbf{M}$ and vector $\mathbf{d}$ are omitted as by now well known in the literature [3], [8], [10].

Definition 2.1: Let $\mathcal{X}_{MCAS}$ be the set of initial states $\mathbf{x}_k$ for which the optimal control problem (5) is feasible (that is the MCAS)

$$\mathcal{X}_{MCAS} = \{\mathbf{x}_k \in \mathbb{R}^{n_x} \mid \exists \underrightarrow{\mathbf{c}}_k \in \mathbb{R}^{n_c n_u}, \mathbf{M}\mathbf{x}_k + \mathbf{N}\underrightarrow{\mathbf{c}}_k \le \mathbf{d}\}.$$

For convenience the same matrices $\mathbf{M}, \mathbf{d}$ as for the MAS have been used, although in practice the minimal forms of these sets allow the MAS to use fewer rows.

Remark 2.1: The Optimal MPC (OMPC) algorithm is given by solving the QP optimisation (5) at every sampling instant then implementing the first component of $\underrightarrow{\mathbf{c}}_k$, that is $\mathbf{c}_k$ in the control law of (4). When the unconstrained control law is not predicted to violate constraints (i.e. $\mathbf{x}_k \in \mathcal{X}_{MAS}$), the optimising $\underrightarrow{\mathbf{c}}_k$ is zero so the control law is $\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k$.

The optimisation (5) can require a large n_c (d.o.f.) to obtain both good performance and a large feasible region.

B. Laguerre polynomials and the underlying dynamic

This section focuses on efficient mechanisms for generating the Laguerre polynomials in a format that is useful for MPC. Laguerre polynomials are defined as follows:

$$\mathbf{L}_i(z) = \sqrt{(1-a^2)} \frac{(z^{-1} - a)^{i-1}}{(1 - az^{-1})^i}; \quad 0 \le a < 1 \tag{6}$$

These have a time constant of 'a' and thus allow for the perturbation signals $\mathbf{c}_k$ that enter the MAS over a slower time scale than single perturbations as in the OMPC algorithm; consequently the associated MCAS may be bigger [12].

The Laguerre sequences can be computed using the following state-space model.

$$\mathbf{L}_{k+1} = \underbrace{\begin{bmatrix} a & 0 & 0 & 0 & \cdots \\ \beta & a & 0 & 0 & \cdots \\ -a\beta & \beta & a & 0 & \cdots \\ a^2\beta & -a\beta & \beta & a & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{\mathbf{A}_L} \mathbf{L}_k;$$

$$\mathbf{L}_0 = \sqrt{1-a^2}\left[1, \ -a, \ a^2, \ -a^3, \ \ldots\right]^T; \quad \beta = 1 - a^2; \tag{7}$$

where $\mathbf{L}_i(z) = \mathbf{e}_i^T[\mathbf{L}_0, \ \mathbf{L}_1, \ \mathbf{L}_2, \ ...][1, \ z^{-1}, \ z^{-2}, \ ...]^T$ and $\mathbf{e}_i$ is the i-th standard basis vector. The dimension of the state-space predictor (7) can be taken as large (or small) as needed to capture the desired polynomial sequences.

C. LOMPC: Laguerre polynomials and MPC

Laguerre OMPC (LOMPC) is a dual mode MPC algorithm [12] where the input predictions are parameterised in terms of Laguerre polynomials (8). First define the input perturbations as follows:

$$\begin{bmatrix} \mathbf{c}_k \\ \vdots \\ \mathbf{c}_{k+n-1} \\ \vdots \end{bmatrix} = \underrightarrow{\mathbf{c}}_k = \underbrace{\begin{bmatrix} \mathbf{L}_0^T \\ \vdots \\ \mathbf{L}_{n-1}^T \\ \vdots \end{bmatrix}}_{\mathbf{H}_L} \underrightarrow{\eta}_k; \tag{8}$$

where $\underrightarrow{\eta}_k$ is the n_η dimensional decision variable when one uses the first n_η columns of $\mathbf{H}_L$.

The predicted cost is represented in terms of perturbations $\mathbf{c}_k$ as $J = \sum_{i=0}^{\infty} \mathbf{c}_{k+i}^T \mathbf{S} \mathbf{c}_{k+i}$; however, from (8) note that $\mathbf{c}_{k+i} = \mathbf{L}_i^T \underrightarrow{\eta}_k$ and from (7) $\mathbf{L}_i = \mathbf{A}_L \mathbf{L}_{i-1}$, hence

$$J_L = \sum_{i=0}^{\infty} \underrightarrow{\eta}_k^T \mathbf{L}_i \mathbf{S} \mathbf{L}_i^T \underrightarrow{\eta}_k = \underrightarrow{\eta}_k^T \left[\sum_{i=0}^{\infty} \mathbf{A}_L^i \mathbf{L}_0 \mathbf{S} \mathbf{L}_0^T (\mathbf{A}_L^i)^T \right] \underrightarrow{\eta}_k \tag{9}$$

Constraints are rewritten as $\mathbf{M}\mathbf{x}_k + \mathbf{N}\mathbf{H}_L \underrightarrow{\eta}_k \le \mathbf{d}$.

Algorithm 2.1: LOMPC

$$\underset{\rightarrow}{\eta}_k^* = \arg\min_{\underset{\rightarrow}{\eta}_k} \quad J_L \quad s.t. \quad \mathbf{M}\mathbf{x}_k + \mathbf{N}\mathbf{H}_L \underset{\rightarrow}{\eta}_k \leq \mathbf{d}; \tag{10}$$

Define $\underset{\rightarrow}{\mathbf{c}}_k^* = \mathbf{H}_L \underset{\rightarrow}{\eta}_k^*$ and implement $\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k + \mathbf{e}_1^T \underset{\rightarrow}{\mathbf{c}}_k^*$.

Remark 2.2: It is straightforward to show, with conventional arguments, that the LOMPC algorithm is guaranteed to give recursive feasibility and stability in the nominal case and offset free tracking whenever the set point is feasible.

Remark 2.3: If LOMPC uses $a = 0$, then $\mathbf{L}_0 = [1, 0, \ldots]$ and $\mathbf{A}_L$ becomes a shift matrix, that is ones on the lower diagonal. In this case, LOMPC is equivalent to OMPC.

Remark 2.4: Although not discussed here to avoid tedious but straightforward algebra, this section can equally be reworked for a multivariable case allowing for a different time constant 'a' in each loop.

III. The best choice for initial condition in the prediction dynamics

The background section has shown two clear choices in the parameterisation of the flexibility within the future input predictions. First one can choose the implied dynamic $\mathbf{A}_L$ which is usually taken to be a shift matrix (ones in the lower diagonal and zero elsewhere) and second, one can choose the initial condition $\mathbf{L}_0$. This section explores the second of these choices by asking what impact this choice has on feasibility and performance?

A. Structure of predictions

Assume for now that the perturbation signal $\underset{\rightarrow}{\mathbf{c}}_k$ can be defined from the dynamic model:

$$\mathbf{L}_{k+1} = \Phi\mathbf{L}_k; \quad \mathbf{L}_0 = \text{ to be determined.} \tag{11}$$

The term $\mathbf{L}_0$ appears to play a key role in the shape of the predictions that are allowed.

B. The interaction between η predictions and $\boldsymbol{L}_0$

It is clear that $\mathbf{L}_0$ appears to influence the shape of the allowable predictions for $\underset{\rightarrow}{\mathbf{c}}_k$, so this section aims to investigate that flexibility more carefully and ask whether the initialization of $\mathbf{L}_0$ can be exploited to improve either feasibility or performance.

Lemma 3.1: The predictions $\mathbf{c}_{k+i|k}$[1] can be considered as the output of a simple state-space model with initial condition linked to $\mathbf{L}_0$.

Proof. Define a state-space model as follows

$$\left.\begin{array}{c} \mathbf{w}_{k+1} = \Phi^T\mathbf{w}_k + \mathbf{B}\mathbf{u}_k; \\ \mathbf{c}_k = \underset{\rightarrow}{\eta}^T\mathbf{w}_k; \end{array}\right\} \quad \left\{\begin{array}{l} \mathbf{w}_0 = \mathbf{L}_0; \\ \mathbf{u}_i = 0, \forall i \geq 0 \end{array}\right. . \tag{12}$$

Forming predictions for this gives $\mathbf{c}_k = \underset{\rightarrow}{\eta}^T\mathbf{L}_0$, $\mathbf{c}_{k+1} = \underset{\rightarrow}{\eta}^T\Phi^T\mathbf{L}_0$, $\mathbf{c}_{k+2} = \underset{\rightarrow}{\eta}^T(\Phi^T)^2\mathbf{L}_0$, $\ldots$ □

Lemma 3.2: If Φ has distinct eigenvalues, then there exists an alternative state-space model with the same dynamics but different initial condition and different output matrix that gives the same output predictions as (12).

[1] $k+i|k$ means the prediction for sample $k+i$ made at sample k.

Proof. In place of initial condition $\mathbf{L}_0$, instead use $\mathbf{Z}_0$ so now define the model.

$$\left.\begin{array}{c} \mathbf{v}_{k+1} = \Phi^T\mathbf{v}_k + \tilde{\mathbf{B}}\mathbf{u}_k; \\ \mathbf{c}_k = \underset{\rightarrow}{\tilde{\eta}}^T\mathbf{v}_k; \end{array}\right\} \quad \left\{\begin{array}{l} \mathbf{v}_0 = \mathbf{Z}_0; \\ \mathbf{u}_i = 0, \forall i \geq 0 \end{array}\right. . \tag{13}$$

The requirement is that the output predictions of models (12) and (13) given next can be made the same but with $\mathbf{Z}_0 \neq \mathbf{L}_0$ and $\underset{\rightarrow}{\eta} \neq \underset{\rightarrow}{\tilde{\eta}}$. Hence, prove

$$\mathbf{c}_{k+n} = \underset{\rightarrow}{\eta}^T(\Phi^T)^n\mathbf{L}_0 = \underset{\rightarrow}{\tilde{\eta}}^T(\Phi^T)^n\mathbf{Z}_0, \ \forall n. \tag{14}$$

First, decompose Φ using its eigenvalue/vector decomposition into the form $\Phi^T = \mathbf{W}\Lambda\mathbf{V}$ and substitute in (14):

$$\mathbf{c}_{k+n} = \underset{\rightarrow}{\eta}^T\mathbf{W}\Lambda^n\mathbf{V}\mathbf{L}_0 = \underset{\rightarrow}{\tilde{\eta}}^T\mathbf{W}\Lambda^n\mathbf{V}\mathbf{Z}_0. \tag{15}$$

From this it is clear that

$$\mathbf{c}_{k+i} = \sum_j \lambda_j^i\alpha_j; \quad \left\{\begin{array}{c} \alpha_j = [\underset{\rightarrow}{\eta}^T\mathbf{W}]_j[\mathbf{V}\mathbf{L}_0]_j \\ \text{or} \\ \alpha_j = [\tilde{\eta}^T\mathbf{W}]_j[\mathbf{V}\mathbf{Z}_0]_j \end{array}\right. \tag{16}$$

Consequently, there always exists a choice of $\underset{\rightarrow}{\tilde{\eta}}$ so that the output of model (13) replicates (12), as long as the initial condition $\mathbf{Z}_0$ is not orthogonal to any eigenvector of Φ. □

However, the reader will notice that the implied state-space matrix with the Laguerre polynomials has repeated eigenvalues and thus does not have a simple decomposition.

Remark 3.1: For dynamic matrices of the form given in (7), numerical examples demonstrate that the initial condition still has no bearing on the reachable space of future $\mathbf{c}_{k+i}$, but a formal proof remains future work. The key difference is that a non-simple Jordan form is required. It is still possible to decompose the matrix as $\Phi = \mathbf{V}\mathbf{J}\mathbf{V}^{-1}$, but as $\mathbf{J}$ is no longer strictly diagonal, so a more elaborate proof is needed.

C. Recursive feasibility

This section demonstrates that the structure of predictions deployed in (11) ensures a recursive feasibility result.

Theorem 3.1: If the values of $\mathbf{c}_k$ are restricted by (11), then nevertheless, it is always possible to choose $\mathbf{c}_{k+i|k+1} = \mathbf{c}_{k+i|k}$.

Proof. In order to make $\mathbf{c}_{k+i|k} = \mathbf{c}_{k+i|k+1}$, $\forall i > 0$ it is sufficient to make $\mathbf{L}_0^T\Phi^i\eta_k = \mathbf{L}_0^T\Phi^{i-1}\eta_{k+1}$. This is easy to do by choosing $\eta_{k+1} = \Phi\eta_k$. □

Remark 3.2: The reader is reminded that using conventional arguments in the literature, recursive feasibility is sufficient to guarantee convergence and stability.

D. Summary

In summary, subject to some very mild conditions on including components of all the eigenvectors, the choice of $\mathbf{L}_0$ has no impact on the reachable space of $\mathbf{c}_{k+i}$ and thus is not a parameter that needs to be considered further except for the relatively minor issue of the possible impact on numerical conditioning. Another key observation is that the structure of the predictions deployed in LOMPC is actually quite generic and there is no need to restrict oneself to Laguerre polynomials. In fact, one could use any Φ for (11) (instead of $\mathbf{A}_L$), although this of course opens up the challenging question of which Φ in general one might wish to choose?

IV. Alternative parameterisations

This section looks at how one might define the matrix $\mathbf{H}_L$ in (8). Although LOMPC assumes this has a specific structure linked to the choice of $\mathbf{A}_L$, in general one could conjecture that such a choice is unnecessarily restrictive, and indeed the insights of parametric solutions [1] make this very clear. Hence, here the philosophy is to allow a total open choice for $\mathbf{H}_L$ and propose one new method for how that choice might be made and what benefits that might bring?

For simplicity of presentation, the algebra here is given for the SISO case, but all the arguments equally apply to the MIMO case.

A. The reachable space for the input predictions

Currently the flexibility within the input predictions is given from $\underset{\rightarrow}{\mathbf{c}}_k = \mathbf{H}_L \underset{\rightarrow}{\eta}_k$; for OMPC $\mathbf{H}_L$ is an identity matrix, for LOMPC it comprises terms based on $\mathbf{A}_L$ but in general this matrix could be something else. The row dimension dictates how far into the future one wishes to use non-zero $\mathbf{c}_k$ and the column dimension dictates how many d.o.f. there are in the optimisation problem. In general the reachable space is a small subset of the $\underset{\rightarrow}{\mathbf{c}}_k$ space because normally $\mathbf{H}_L$ is tall and thin, i.e. the output predictions are considered over horizons far greater than the number of d.o.f. in the optimisation.

Assume that the horizon for $\mathbf{c}_{k+i} \neq 0$ (the horizon dimension of $\mathbf{H}_L$) is n_{c2} and the number of free moves[2] is n_c (or n_η). How might one determine the best n_c search directions for $\underset{\rightarrow}{\mathbf{c}}_k$? With LOMPC the search directions are simply defined by the columns of $\mathbf{H}_L$ but it may be the case that these do not include a better search direction.

B. Monte Carlo approaches to find the search directions

In practice it is known from parametric insights that the optimal $\underset{\rightarrow}{\mathbf{c}}_k$ is highly nonlinear in terms of its dependence upon the current state $\mathbf{x}_k$; moreover the complexity of the parametric solution is to some extent unpredictable. Here, a simple Monte Carlo approach is taken.

Algorithm 4.1: Optimal search directions

1) Choose a large value of n_{c2} for OMPC consistent with finding a large enough feasible region.
2) Define equi-spaced points on a retangular grid in the state-space on the outside of an n_x dimensional cube (USER to define desired spacing). Stretch these directions to the boundary of the MCAS (for given large $n_c = n_{c2}$) and denote them as v_i.
3) For each point v_i, determine the optimal $\underset{\rightarrow}{\mathbf{c}}_k$ with $n_c = n_{c2}$ and denote as $\underset{\rightarrow}{\mathbf{c}}_i$.
4) Form a matrix $\mathbf{P} = [\underset{\rightarrow}{\mathbf{c}}_1, \underset{\rightarrow}{\mathbf{c}}_2, \cdots, \underset{\rightarrow}{\mathbf{c}}_n]$.
5) Find the singular value decomposition of $\mathbf{P}$ as $\mathbf{P} = \mathbf{X}\Sigma\mathbf{Y}^*$.
6) Define $\mathbf{H}_L = \mathbf{X}(:, 1 : n_c)$ where $n_c = n_\eta$ is now taken to be the desired number of d.o.f. It is assumed that the first n_c columns of $\mathbf{X}$ correspond to the largest singular values.

[2]Remember that the number of d.o.f. of OMPC is actually $n_c n_u$

It is clear that in some objective sense, this choice of $\mathbf{H}_L$ captures the best finite number of search directions, on average, to capture the optimal $\underset{\rightarrow}{\mathbf{c}}_k$ required on the boundary of the MCAS. Moreover, one can inspect the singular values in Σ to determine what might be the best number of columns to take.

C. Feasible OMPC

This section shows briefly how the $\mathbf{H}_L$ matrix of the previous subsection is used to define a Feasible OMPC algorithm or FOMPC.

Algorithm 4.2: FOMPC

1) Select the maximum number n_{c2} of non-zero $\mathbf{c}_{k+i}$ terms in the predictions and the number of d.o.f. n_η to be used online; use Algorithm 4.1 to determine $\mathbf{H}_L$.
2) Define $J_F = \underset{\rightarrow}{\eta}_k^T \mathbf{S}_L \underset{\rightarrow}{\eta}_k$, where
$$\mathbf{S}_L = \mathbf{H}_L^T \text{diag}\{\mathbf{S}, \ldots, \mathbf{S}\}\mathbf{H}_L$$
3) Find the optimum $\underset{\rightarrow}{\eta}_k^*$ from:
$$\underset{\rightarrow}{\eta}_k^* = \arg\min_{\underset{\rightarrow}{\eta}} \quad J_F \quad s.t. \quad \mathbf{M}\mathbf{x}_k + \mathbf{N}\mathbf{H}_L \underset{\rightarrow}{\eta}_k \leq \mathbf{d}$$
4) Define $\underset{\rightarrow}{\mathbf{c}}_k^* = \mathbf{H}_L \underset{\rightarrow}{\eta}_k^*$ and implement the control law $\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k + \mathbf{e}_1^T \underset{\rightarrow}{\mathbf{c}}_k^*$.

D. Recursive feasibility and convergence

Here, by allowing $\mathbf{H}_L$ to have a totally open structure, the specific structure of (8) is lost and with it the recursive feasibilty result given in Section III-C. A simple procedure does exist in the literature to recover this guarantee, but at the cost of introducing an extra degree of freedom [9]. In simple terms, one appends the d.o.f. in $\mathbf{H}_L \underset{\rightarrow}{\eta}_k$ with one additional direction, that is the tail of the optimised $\underset{\rightarrow}{\mathbf{c}}_k$ from the previous sample. This is not discussed further as by now well known.

V. Numerical examples

This section gives some numerical illustrations to compare the efficacy of OMPC, LOMPC and FOMPC. Specifically the focus is on the comparison of the volume of the $\mathcal{X}_{MCAS}$ against the number of optimisation variables. That is, it considers to what extent LOMPC and FOMPC use a more systematic parameterisation of the d.o.f. within the predictions to allow for maximal gains in feasibility with small numbers of d.o.f.

Two examples with different state dimensions are presented, details in the appendix. A large number of equi-spaced (by solid angle) directions are chosen in the state-space. For each direction, the distance from the origin to the boundary of the MCAS is determined and clearly the larger the distance, hereafter denoted as radius, the better the feasibility. Finally radii are normalised against the radii obtained with OMPC with $n_c = 20$, we realise this is somewhat arbitrary but it seems a pragamatic limit for the global feasible region with sensible sampling and dynamics. In fact it is well known that often there is no benefit taking

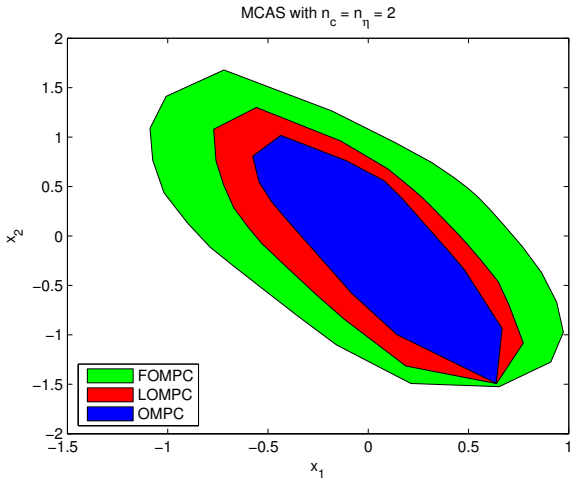

Fig. 1. Comparison MCAS for $n_c = n_\eta = 2$ – Example 1.

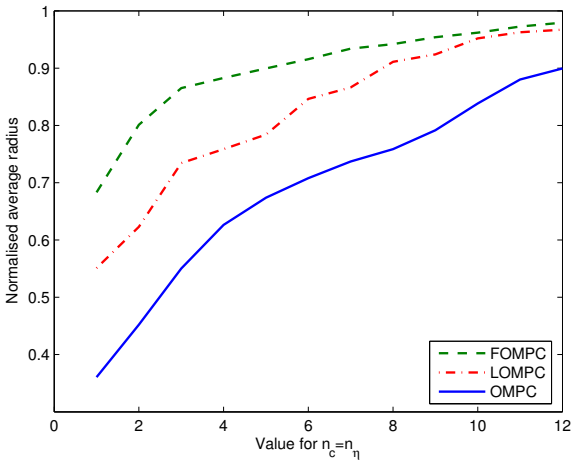

Fig. 2. Comparison of average MCAS radii as n_c, n_η vary compared to MCAS radii with $n_c = 20$ for all three algorithms. Example 1.

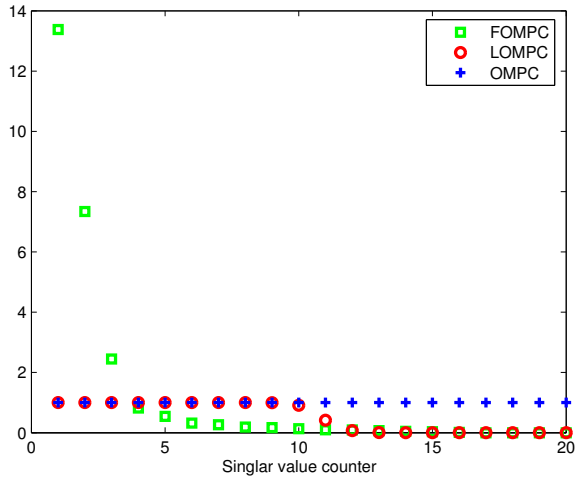

Fig. 3. Comparison of singular values of $\mathbf{H}_L$ – Example 1.

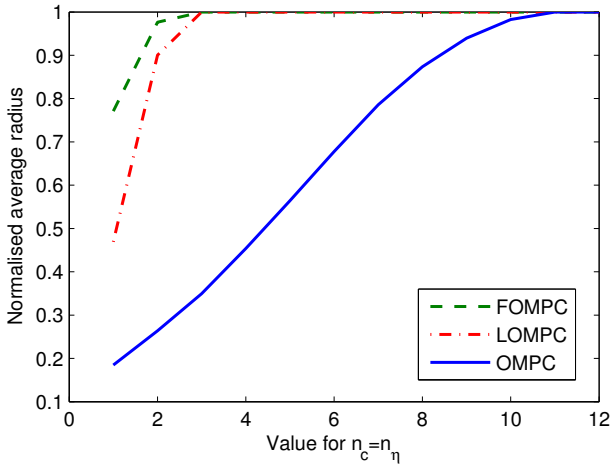

Fig. 4. Comparison of average MCAS radii as n_c, n_η vary compared to MCAS radii with $n_c = 20$ for all three algorithms. Example 2.

n_c beyond a certain value [4] that is typically less than 20. For all examples, the main Laguerre parameter is taken as $a = 0.8$.

The results for the Example 1 are shown in Figures 1-3. From Figure 1, it is clear that FOMPC has a larger MCAS than both LOMPC and OMPC for the same number of d.o.f. Moreover, Figure 2 shows that, for this example, FOMPC gets to within 90% of the global MCAS with just 3 d.o.f. whereas, LOMPC requires about 8 d.o.f. and OMPC requires 12 d.o.f. This message can be reinforced by consideration of the singular values for the respective $\mathbf{H}_L$ matrices shown in Figure 3; clearly the FOMPC algorithm has some large singular values and then many less important directions, the LOMPC algorithm has many equal singular values and then directions become less significant beyond the 9th (probably due to row truncation) and OMPC has equal singular values.

For Example 2, Figure 4 shows that the radii increase as the number of d.o.f. increase. For $n_\eta \geq 3$, both FOMPC and LOMPC reach, to within less than 1%, the MCAS for OMPC with $n_c = 20$.

Remark 5.1: These examples have assumed that n_c, n_η are integers denoting the number of free variables in the future sequences of either $\mathbf{c}$ or the number of Laguerre polynomials. However, in example 2 the number of d.o.f. is larger as it is scaled by the input dimension n_u. Therefore, usually the number of d.o.f. increases in multiples of n_u; although one can have different horizons for different loops it is not immediately obvious how to use this flexibility.

An interesting observation is that the FOMPC approach identifies the best sequence directions and therefore it is no longer necessary or even logical to go up in steps of n_u. Figure 5 shows how the FOMPC algorithm can increase the number of variables, systematically, one at a time and gain feasibility improvements as it goes. In this case the singular values of $\mathbf{H}_L$ reduce as $[1300, 444, 153, 54, 48, 37, ...]$.

VI. Conclusions and Future Work

This paper has investigated different ways of parameterising the d.o.f. within the predictions for an OMPC type of algorithm. The examples make it clear that using just the individual values for future inputs in the near horizon is not efficient in general, even when the terminal control

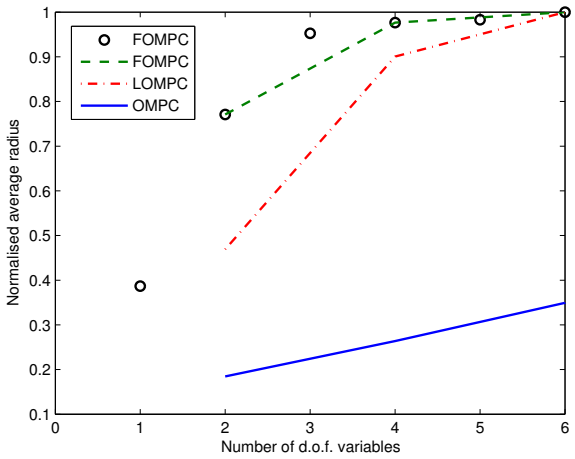

Fig. 5. Comparison of average relative radii of MCAS as total number of d.o.f. vary for all three algorithms. Example 2.

law is embedded within the predictions. Two alternatives are proposed. It is shown that basing future input flexibility in terms of an underlying dynamic model whose order is in effect the number of d.o.f., allows straightforward recursive feasibility statements. A systematic choice for this dynamic remains an open question although numerical evidence has shown that Laguerre polynomials seem to perform very well. A second proposal considers how one might define a more objective definition of best parameterisation and proposes a mechanism based on a Monte Carlo approach, that is find the best sequences for all possible directions and then determine which subset best approximates these. Examples show that the latter approach does indeed give better feasibility for low numbers of d.o.f.

A. Future work

The proposed FOMPC does present a drawback: while it generates a large feasible set for a low number of d.o.f., it produces sub-optimal performance. Future work will try to investigate a way to extend FOMPC in order to provide a good trade-off between the resulting feasible region and performance. The main idea will be to specify a desired radius α and a level of performance loss β. It then can be chosen a set of points $\mathbf{v}$ on the boundary of the $\mathcal{X}_{MCAS}$ and required that $\alpha\mathbf{v}$ is contained in $\text{proj}_x\mathcal{X}_{MCAS}$ and that there exists a point $\mathbf{u}_i$ for each $\mathbf{x}_i$ in $\alpha\mathbf{v}$ such that the cost $J_{\mathbf{x}_i,\mathbf{u}_i}(1+\beta) \leq J^*_{\mathbf{x}_i}$. The key point is that the optimisation problem then will minimise the dimension of the subspace that the points $\mathbf{u}_i$ lie within (i.e. the d.o.f.). This rank minimization is non-convex, but it can instead be minimised the nuclear norm, or sum of singular values, of $[\mathbf{u}_0, \ldots, \mathbf{u}_n]$, which is the convex envelope of the rank (i.e. the closest convex function to the rank function). The result of this approximation is a convex semi-definite program (SDP) that will compute the smallest d.o.f. parameterisation for a desired performance and radius.

Finally, there is interest in asking whether FOMPC can be combined with parametric type thinking, that is to use more directional information into the algorithm and thus improve either or both efficiency and performance.

REFERENCES

[1] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[2] E.F. Camacho and C. Bordons. *Model predictive control*. Springer, 2004.

[3] 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. on Aut. Control*, 36(9):1008–1020, 1991.

[4] P. Grieder, F. Borrelli, F. Torrisi, and M. Morari. Computation of the constrained infinite time linear quadratic regulator. *Automatica*, 40(4):701–708, 2004.

[5] L. Imsland, J.A. Rossiter, B. Pluymers, and J. Suykens. Robust triple mode MPC. *International Journal of Control*, 81(4):679–689, 2008.

[6] M. Kothare, V. Balakrishnan, and M. Morari. Robust constrained predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.

[7] J.M. Maciejowski. *Predictive control with constraints*. Prentice Hall, 2002.

[8] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and PO Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[9] J.A. Mendez, B. Kouvaritakis, and J.A. Rossiter. State-space approach to interpolation in MPC. *International Journal of Robust and Nonlinear Control*, 10(1):27–38, 2000.

[10] J.A. Rossiter. *Model-based predictive control, a practical approach*. CRC Press, London, 2003.

[11] J.A. Rossiter and P. Grieder. Using interpolation to improve efficiency of multiparametric predictive control. *Automatica*, 41(4):637–643, 2005.

[12] J.A. Rossiter, L. Wang, and G. Valencia-Palomo. Efficient algorithms for trading off feasibility and performance in predictive control. *International Journal of Control*, 83(4):789–797, 2010.

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

[14] L. Wang. Discrete model predictive controller design using Laguerre functions. *Journal of Process Control*, 14(2):131–142, 2004.

APPENDIX

Example 1

$$\mathbf{A} = \begin{bmatrix} 0.6 & -0.4 \\ 1 & 1.4 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0.2 \\ 0.05 \end{bmatrix}; \quad \mathbf{C} = \begin{bmatrix} 1 & -2.2 \end{bmatrix};$$
$$-1.5 \leq \mathbf{u}_k \leq 0.8; \ |\Delta\mathbf{u}_k| \leq 0.4; \ |\mathbf{x}_{i,k}| \leq 5; \ \mathbf{Q} = \mathbf{I}; \ \mathbf{R} = 2.$$

Example 2

$$\mathbf{A} = \begin{bmatrix} 0.9146 & 0 & 0.0405 & 0.1 \\ 0.1665 & 0.1353 & 0.0058 & -0.2 \\ 0 & 0 & 0.1353 & 0.5 \\ -0.2 & 0 & 0 & 0.8 \end{bmatrix};$$

$$\mathbf{B} = \begin{bmatrix} 0.054 & -0.075 \\ 0.005 & 0.0147 \\ 0.864 & 0 \\ 0.5 & 0.2 \end{bmatrix}; \mathbf{C} = \begin{bmatrix} 1.799 & 13.21 & 0 & 0.1 \\ 0.823 & 0 & 0 & -0.3 \end{bmatrix};$$

$$|\Delta\mathbf{u}_{i,k}| \leq 2; \ |\mathbf{u}_k| \leq \begin{bmatrix} 1 \\ 2 \end{bmatrix}; \ |\mathbf{x}_{i,k}| \leq 10; \ \mathbf{Q} = \mathbf{C}^T\mathbf{C}; \ \mathbf{R} = \mathbf{I}.$$