Proceedings Paper:

Rossiter, J.A. orcid.org/0000-0002-1336-0633, Gonzalez-Villareal, O.J. and Sarbini, M.A.M. (2023) A review of parameterised MPC algorithms. In: Ishii, H., Ebihara, Y., Imura, J. and Yamakita, M., (eds.) IFAC-PapersOnLine. 22nd World Congress of the International Federation of Automatic Control (IFAC2023), 09-14 Jul 2023, Yokohama, Japan. Elsevier , pp. 7692-7697.

https://doi.org/10.1016/j.ifacol.2023.10.1171

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 7692–7697

A Review of Parameterised MPC Algorithms ⋆

John A. Rossiter * **Oscar J. Gonzalez-Villareal** * **Mohammad A. M. Sarbini** **

* *Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, UK (e-mail: j.a.rossiter@sheffield.ac.uk)*
** *Electrical and Electronic Engineering Programme Area, Universiti Teknologi Brunei (email: adi.sarbini@utb.edu.bn)*

Abstract: This paper provides an evaluation and comparison of popular parameterised model predictive control approaches that have been proposed in the literature in recent years. Using the Generalised Predictive Control (GPC) algorithm as the baseline algorithm, the paper sets out a number of performance criteria to compare and contrast with several other MPC approaches. Numerical examples use 100 random samples of 2, 3, and 4-state models and the approaches are compared using the selected performance criteria.

Keywords: Predictive control, Model predictive and optimization-based control, Monte Carlo methods

1. INTRODUCTION

This paper takes as a start point the huge success of MPC (Qin and Badgwell, 2003; García et al., 1989) in industry and the corresponding huge interest in the academic literature. From early beginnings (Cutler and Ramaker, 1980; Clarke et al., 1987), through a growing understanding of how to ensure stability with terminal modes (Rawlings, 2000; Rossiter et al., 1998; Mayne et al., 2000) the literature is now awash with papers covering far more demanding issues, such as robustness (Kothare et al., 1996), feasibility (Chisci et al., 2001; Mayne et al., 2005), non-linear methods (Henson, 1998; Grüne and Pannek, 2017), computational methods and much more.

As the literature focuses on more challenging problems and scenarios it is unsurprising that the computational demands grow and thus there is a significant interest in efficient optimisation of MPC-related problems. However, many of these papers resort to the classical assumption that the degrees of freedom are the individual values of the future inputs. Conversely, a few papers have asked the question: would a re-parameterisation of the degrees of freedom be useful (Cagienard et al., 2007; Abdullah and Rossiter, 2021; Wang, 2009; Khan and Rossiter, 2013) Indeed this concept is implicit in reference governor approaches (Garone et al., 2017) which use the input to the loop as the degree of freedom rather than the input itself and those are known to allow some computational simplification.

The literature lacks a proper overview of the various parameterisation approaches and some form of comparison between them. When and why would you use a specific parameterisation? Consequently, the main contribution of this paper is to provide such a survey and to set out a number of criteria by which these methods can be evaluated. In order not to over-complicate the paper, the focus here will solely be on the linear case. Future work will consider the differences when one extends these concepts to the non-linear case.

Section 2 makes arguments for a workable set of criteria for comparing different MPC algorithms. Section 3 introduces several input parameterisations and the associated MPC algorithms. Section 4 presents numerical comparisons and the paper then finishes with conclusions.

2. OVERVIEW OF PERFORMANCE CRITERIA

This section sets out a number of criteria which can be used to compare and contrast the various MPC algorithms to be considered. Nevertheless, it is important to note that, to some extent, any such criteria are arbitrary; there is always some subjectivity in deciding how *best* will be defined. It is reasonable to draw figures, as in multi-objective optimisation (Ishibuchi, 1995), showing how performance varies against different criteria with different choices.

2.1 System definition

For convenience hereafter, and without loss of generality, take the following *nominal* square state-space model:

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

state $\mathbf{x}_k$, output $\mathbf{y}_k$, input $\mathbf{u}_k$ and output disturbance $\mathbf{d}_k$, input/output dimension N and state dimension N_x.

2.2 Performance index measures

A typical performance index J, for a constant target $\mathbf{r}$ and $\mathbf{u}_{ss}, \mathbf{x}_{ss}$ the expected steady-state input/state, is:

⋆ This work is sponsored by the Government of Brunei Darussalam.

10.1016/j.ifacol.2023.10.1171

$$J = \sum_{i=1}^{n_y} \|\mathbf{r} - \mathbf{y}_{k+i}\|_{Q_y}^2 + \sum_{i=1}^{N_u} \|\mathbf{u}_{k+i-1} - \mathbf{u}_{\rm ss}\|_R^2 \\ \equiv \sum_{i=1}^{n_y} \|\mathbf{x}_{\rm ss} - \mathbf{x}_{k+i}\|_{Q}^2 + \sum_{i=1}^{N_u} \|\mathbf{u}_{k+i-1} - \mathbf{u}_{\rm ss}\|_R^2 \quad (2)$$

where the notation $\|\mathbf{x}\|_Q^2 = \mathbf{x}^T Q \mathbf{x}$ and n_y, N_u are the output and input prediction horizons.

2.3 GPC/DMC (Clarke et al., 1987): a default algorithm

Predictions over a finite horizon for model (1) can be expressed in the following format (Rossiter, 2018).

$$\mathbf{Y} = P_x \mathbf{x}_k + H\mathbf{U}_k \quad (3)$$

for suitable P_x, H and $\mathbf{Y} = [\mathbf{y}_{k+1}^T, \mathbf{y}_{k+2}^T, \cdots, \mathbf{y}_{k+n_y}^T]^T$, $\mathbf{U}_k = [\mathbf{u}_k^T, \mathbf{u}_{k+1}^T, \cdots, \mathbf{u}_{k+n_u-1}^T]^T$; assume $\mathbf{u}_{k+n_u+i} = \mathbf{u}_{k+n_u-1}, \forall i \geq 0$. Substitution of predictions (3) into performance index (2), with $n_u = N_u$, results in:

$$J = \mathbf{U}_k^T \underbrace{[H^T Q H + R]}_{S} \mathbf{U}_k + \mathbf{U}_k^T \underbrace{[2H^T P_x]}_{P} \mathbf{x}_k + \alpha_k \quad (4)$$

where α_k does not depend on $\mathbf{U}_k$. So, in a more generic format the required MPC optimisation reduces to:

$$\min_{\mathbf{w}} \quad J = \mathbf{w}^T S \mathbf{w} + \mathbf{w}^T P \mathbf{x}_k \quad (5)$$

where here the substitution has been made that $\mathbf{w} = \mathbf{U}_k$ and matrices S, P are the core quadratic parameters which impact on the conditioning of the optimisation.

2.4 Constraints and feasible regions

Most practical systems have constraints on inputs, states and outputs, for example, with $\Delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$:

$$\left.\begin{array}{ll} \underline{u} \leq \mathbf{u}_k \leq \overline{u}; & \underline{\Delta u} \leq \Delta\mathbf{u}_k \leq \overline{\Delta u} \\ \underline{x} \leq \mathbf{x}_k \leq \overline{x}; & \underline{y} \leq \mathbf{y}_k \leq \overline{y} \end{array}\right\} \forall k \geq 0 \quad (6)$$

More nuanced constraints not included here for simplicity.

We need to ensure that the predictions (3) for system (1) do not violate constraints (6) for all future time. Assuming we have sufficient asymptotic information on the future inputs $\mathbf{u}_k$, this reduces to a set of linear matrix inequalities:

$$\mathcal{N}\mathbf{U}_k + \mathcal{M}\mathbf{x}_k + \mathcal{L}\mathbf{u}_{k-1} + \mathcal{G}\mathbf{r}_{k+1} + \mathcal{T}\mathbf{d}_k \leq \mathbf{f} \quad (7)$$

for appropriate matrices $\mathcal{M}, \mathcal{N}, \mathcal{L}, \mathcal{G}, \mathcal{T}, \mathbf{f}$ (e.g. (Gilbert and Tan, 1991; Boyd et al., 1994; Blanchini and Miani, 2015)). This paper does not consider efficient computations (Pluymers et al., 2005).

One of the main aims of this paper is to evaluate different MPC control designs and specifically, one core measure is the volume, in $\mathbf{x}-$space of the set associated to (7). Ideally we want to know that we can satisfy constraints with the allowable choices of $\mathbf{U}_k$ for as large a region of states as possible. So, defining the feasible region as $\mathcal{S}$:

$$\mathcal{S} = \{\, \mathbf{x} : \exists \mathbf{U_k} \ \text{ s.t. } \ (7) \,\} \quad (8)$$

In practice the feasible region $\mathcal{S}$ depends on a number of time varying values $(\mathbf{u}_{k-1}, \mathbf{r}_k, \mathbf{d}_k)$, hence there is no simple unique definition for the feasible region (Rossiter and Dughman, 2018).

Remark 1. It is well known that the target value $\mathbf{r}_k$ can be a very useful tool for increasing feasible volumes (Rossiter, 2006) and indeed is an underlying motivation for reference governor approaches (Garone et al., 2017) but that is outside the remit of this paper.

One can argue that many industrial systems have a common steady-state which, using deviation variables, can be represented as the origin. Thus we define all variables as deviations about the target point, $\mathbf{r}_k = 0$. Similarly, assume that simulations start from a steady-state such as the origin so all subsequent states ultimately came from the origin. This allows a weak assumption that assuming $\mathbf{u}_{k-1} = 0$ will not impact significantly on the *desired* volume comparisons for $\mathcal{S}$; moreover the impact of $\mathbf{u}_{k-1} = 0$ is likely to be similar across many algorithms. In a similar way, while disturbances $\mathbf{d}_k$ do affect the feasible region (Rossiter, 2006; Rossiter and Dughman, 2018), this impact is expected to be consistent across different parameterisations of the future inputs $\mathbf{U}_k$, so, for a pragmatic comparison, it is judicious to ignore this effect. In summary, this paper will focus its comparison on the following set definition.

$$\mathcal{S} = \{\, \mathbf{x} : \ \ \exists \mathbf{U}_k \ \text{ s.t. } \ \mathcal{N}\mathbf{U}_k + \mathcal{M}\mathbf{x} \leq \mathbf{f} \,\} \quad (9)$$

The expectation is that the comparative volumes for region (9) across different MPC algorithms will be similar to those achieved with the time varying regions defined in eqn. (8) and thus enable a useful comparison.

2.5 Numerical conditioning

Although the optimisation complexity may be similar, the optimisation conditioning may not be. It has long since been known (Rossiter et al., 1998) that a re-parameterisation of the d.o.f. can have a significant and sometimes critical impact on the numerical conditioning, especially for systems with open-loop unstable poles. This impact would be even more important for processors using fewer bits (lower accuracy).

Poor conditioning will predominantly enter through the required quadratic programming problem which will reduce to a structure of the following form:

$$\mathbf{w}^T S \mathbf{w} + \mathbf{w}^T P \mathbf{x} \ \text{ s.t. } \ \mathcal{N}\mathbf{w} + \mathcal{M}\mathbf{x} \leq \mathbf{f} \quad (10)$$

One could argue that the conditioning of all these matrices (S, P, N, M) will be important (and probably similar as based on the same predictions), but it is likely that matrix S will be most important as this is the one that needs inverting and, being square and dependent on the step response matrix, will capture the poor conditioning present in all the other matrices. In summary, a useful comparison criteria will be the condition number of the matrix S in (10).

3. POPULAR INPUT PARAMETERISATIONS WITHIN MPC

The main purpose of this paper is to explore whether the choice of input parameterisation within the predictions has a meaningful and useful impact on the various performance criteria outlined in Section 2. If there is strong evidence of this impact, then this choice becomes an important design decision early in the control design process. This section gives a concise introduction to a number of possible input parameterisations which have appeared in the literature with a focus on those which are more common.

Remark 2. For convenience the number of sample d.o.f. is taken to be n_u, that is, over how many samples are the d.o.f. distributed? The total number of d.o.f. is Nn_u.

3.1 Using future input values directly in the open-loop

From the early days of MPC (Clarke et al., 1987; Cutler and Ramaker, 1980) it was common to define the degrees of freedom (d.o.f.) $\mathbf{w}$ within the predictions as follows:

$$\mathbf{w} = \begin{bmatrix} \Delta\mathbf{u}_k \\ \Delta\mathbf{u}_{k+1} \\ \vdots \\ \Delta\mathbf{u}_{k+n_u-1} \end{bmatrix}; \quad \{\Delta\mathbf{u}_{k+n_u+i} = 0, \quad \forall i \geq 0\} \tag{11}$$

An equivalent parameterisation would be to use absolute values, with $\mathbf{u}_k$ and assumed to be constant after n_u steps.

This choice is summarised in the constraints and cost functions of (9), (10) (albeit there would be minor differences depending on the choice of $\mathbf{w}_k$ in (11); rates or absolute values). Hence a core point is that:

$$S = H^T Q H + R \tag{12}$$

Remark 3. With only input constraints and no input rate constraints, the inequalities reduce to an Nn_u dimensional cuboid, which tailored QP algorithms can handle very efficiently; this is discussed no further in this paper.

3.2 Using Laguerre Polynomials (LMPC)

An obvious weakness of (11) is the implied terminal constraint that the input becomes constant after n_u steps. It is easy to find examples where that is far from the expected or most desired input trajectory (Rossiter, 2018). Consequently, some authors have considered the potential benefit of building up the future input trajectories as a combination of functions (Khan and Rossiter, 2013; Wang, 2004; Valencia-Palomo and Rossiter, 2012). The most common choice seems to be Laguerre functions and there are expected conditioning and computational advantages in using orthonormal functions.

Assume the Laguerre polynomials $L_i(z)$ are defined as:

$$L_i(z) = \sum_{k=0}^{\infty} l_{i,k} z^{-k} \tag{13}$$

The future inputs, for an infinite prediction horizon, are defined using a linear combination of the Laguerre polynomials, for example (using SISO case to simplify presentation):

$$\mathbf{U}_k = \underbrace{\begin{bmatrix} l_{1,k+1} & l_{2,k+1} & l_{3,k+1} & \cdots \\ l_{1,k+2} & l_{2,k+2} & l_{3,k+2} & \cdots \\ l_{1,k+3} & l_{2,k+3} & l_{3,k+3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{F} \mathbf{w} \tag{14}$$

The matrix in (14) is somewhat awkward to work with, hence it is easier to recognise that, assuming one needs just the first m functions, an equivalent form exists (Wang, 2004) which is much more amenable to the algebra required for MPC:

$$\begin{bmatrix} l_{1,k+1} \\ l_{2,k+1} \\ \vdots \\ l_{m,k+1} \end{bmatrix} = \Psi \begin{bmatrix} l_{1,k} \\ l_{2,k} \\ \vdots \\ l_{m,k} \end{bmatrix} \tag{15}$$

where Ψ is a suitable $m \times m$ matrix and similarly with $[l_{1,0} \ \cdots \ l_{m,0}]^T = L_0$ appropriately defined. Consequently, after some small manipulation, one can find a generic and compact input prediction representation of (14) using:

$$\mathbf{u}_{k+i} = L_0^T (\Psi^T)^i \mathbf{w} \implies \mathbf{u}_k = F\mathbf{w} \tag{16}$$

It is noted that the formulation of the input predictions using a state space model can enable convenient and fast computations (Khan and Rossiter, 2013) for output predictions, constraints and cost functions. Indeed, one advantage of this latter form over (14) is that it works much better with infinite horizon performance indices and admissible sets, as one can exploit Lyapunov equations rather than explicit enumeration.

Constraint inequalities with Laguerre MPC. At first sight one might assume that the constraints can be assessed using (7) directly, so substituting in from (14):

$$\mathcal{N} F\mathbf{w} + \mathcal{M}\mathbf{x} \leq \mathbf{f} \tag{17}$$

Optimisation problem with Laguerre MPC This formulation of LMPC is analogous to GPC with the only obvious difference being that, implicitly, the control horizon implicit in (14) is taken to be much longer than the number of sample d.o.f. n_u; logically one could take $N_u = n_y$. Hence the predictions would be similar to (3):

$$\mathbf{Y}_k = P_x \mathbf{x}_k + HF\mathbf{w}_k; \quad F\mathbf{w}_k = \mathbf{U}_k \tag{18}$$

Substitution of predictions (18) into performance index (2):

$$\begin{aligned} J = \mathbf{w}_k^T & \underbrace{[F^T H^T Q H F + F^T R F]}_{S} \mathbf{w}_k \\ & + \mathbf{w}_k^T \underbrace{[2F^T H^T P_x]}_{P} \mathbf{x}_k + \alpha_k \end{aligned} \tag{19}$$

where α_k does not depend on $\mathbf{w}_k$ and we re-emphasise that the H matrix here will be square rather than tall and thin (as in GPC). The core point here is that the matrix to be checked for conditioning is given as:

$$S = F^T H^T Q H F + F^T R F \tag{20}$$

3.3 Input blocking MPC (IBMPC)

Input blocking (Cagienard et al., 2007) is another method for extending the temporal impact of the d.o.f. while keeping the dimension of the d.o.f. vector low. One assumes that the input holds its value for a number of samples between changes, so for example (using a block size of $b = 2$ for illustration):

$$\underset{(k \text{ even})}{\mathbf{U}_k} = \begin{bmatrix} \mathbf{v}_k \\ \mathbf{v}_k \\ \hline \mathbf{v}_{k+2} \\ \mathbf{v}_{k+2} \\ \hline \vdots \\ \hline \mathbf{v}_{k+m-1} \\ \mathbf{v}_{k+m-1} \\ \hline \mathbf{v}_{k+m} \end{bmatrix}; \quad \underset{(k \text{ odd})}{\mathbf{U}_k} = \begin{bmatrix} \mathbf{v}_k \\ \hline \mathbf{v}_{k+1} \\ \mathbf{v}_{k+1} \\ \hline \mathbf{v}_{k+2} \\ \mathbf{v}_{k+2} \\ \hline \vdots \\ \hline \mathbf{v}_{k+m} \\ \mathbf{v}_{k+m} \end{bmatrix}; \tag{21}$$

with $\{\mathbf{u}_{k+mb+i} = \mathbf{v}_{k+m+i} = \mathbf{v}_{k+m}, \ \forall i \geq 0\}$ and the d.o.f. are $\mathbf{w} = [\mathbf{v}_k^T, \cdots, \mathbf{v}_{k+m}^T]^T$.

Remark 4. The *sample* dimension of $\mathbf{U}_k$ is taken as $mb+1$ and to ensure recursive feasibility, $\mathbf{U}_k$ has a periodic time varying structure, the period being equal to the block size

b. It is straightforward to define matrices $M_j,\ j = 1, ..., b$ to give the relationship between d.o.f. $\mathbf{w}$ and $\mathbf{U}_k$, e.g.

$$\begin{array}{l} \{\mathbf{U}_k = M_1\mathbf{w}_k; \quad k = 0, b, 2b, ...\}; \\ \{\mathbf{U}_k = M_2\mathbf{w}_k; \quad k = 1, b+1, 2b+1, ...\}; \\ \quad\vdots \\ \{\mathbf{U}_k = M_b\mathbf{w}_k; \quad k = b-1, 2b-1, 3b-1, ...\}; \end{array} \tag{22}$$

Constraints with IBMPC The constraint inequalities exploit (22) with (7) and thus take the following periodic time varying form to match the definition of $\mathbf{U}_k$:

$$\mathcal{N}M_j\mathbf{w} + \mathcal{M}\mathbf{x} \le \mathbf{f}; \quad j = 1, 2, \cdots, b, 1, \cdots, b, 1, \cdots \tag{23}$$

That is, j will cycle. In terms of numbers of inequalities/rows in the constraints (23), for the same dimension of $\mathbf{w}$, this will be same as for GPC/DMC.

Performance index with IBMPC This formulation is closely analogous to subsection 3.2.2 due to similarities between the input parameterisations of (14) and (22). The only significant difference is that the performance index is periodically time varying, with period b. The predictions are given as (with the appropriate columns of H):

$$\mathbf{Y}_k = P_x\mathbf{x}_k + HM_j\mathbf{w}_k; \quad M_j\mathbf{w}_k = \mathbf{U}_k; \quad j = 1, \cdots b \tag{24}$$

Substitution of predictions (24) into (2) yields:

$$\begin{array}{l} J_j = \mathbf{w}_k^T \underbrace{[H^T M_j^T Q M_j H + M_j^T R M_j]}_{S_j} \mathbf{w}_k \\ \quad + \mathbf{w}_k^T \underbrace{[2M_j^T H^T P_x]}_{P_j} \mathbf{x}_k + \alpha_k; \quad j = 1, ..., b \end{array} \tag{25}$$

where α_k does not depend on $\mathbf{w}_k$ and the performance index is periodically time varying with period b. The core parameter S (or S_j) is time varying:

$$S_j = [H^T M_j^T Q M_j H + M_j^T R M_j], \quad j = 1, \cdots, b \tag{26}$$

3.4 Dual mode MPC (DMPC)

Dual-mode predictions (Clarke and Scattolini, 1991; Kouvaritakis et al., 1992; Rawlings and Muske, 1993; Scokaert and Rawlings, 1998; Mayne et al., 2000) stem from a similar motivation as the use of Laguerre functions, that is, how do we embed a sensible long term dynamic into the input predictions, but without requiring a large number of d.o.f.? A simple answer is to also ask; *what is a sensible unconstrained control law?* Then, base long term predictions on this control law, assuming that no constraints are violated. Here we use a simple state feedback $\mathbf{u}_k = -K\mathbf{x}_k$ to represent this unconstrained control law (Scokaert and Rawlings, 1998) and omit details linked to embedding integral action. The d.o.f. are parameterised as deviations from the unconstrained control law. Hence, the proposed dual mode structure (Rossiter et al., 1998) takes the following form, to be considered alongside model (1):

$$\mathbf{U}_k = \underbrace{\begin{bmatrix} -K\mathbf{x}_k \\ -K\mathbf{x}_{k+1} \\ \vdots \\ -K\mathbf{x}_{k+n_u-1} \\ \hline -K\mathbf{x}_{k+n_u} \\ \vdots \end{bmatrix}}_{\overline{\mathbf{U}}} + \underbrace{\begin{bmatrix} I \\ 0 \end{bmatrix}}_{F_D} \underbrace{\begin{bmatrix} \mathbf{c}_k \\ \mathbf{c}_{k+1} \\ \vdots \\ \mathbf{c}_{k+n_u-1} \end{bmatrix}}_{\mathbf{w}} \tag{27}$$

or more generally $\mathbf{U}_k = \overline{\mathbf{U}} + F_D\mathbf{w}_k$. In (27) the first n_u predicted values of the input constitute the d.o.f. $\mathbf{w}_k$.

Constraint handling with DMPC In terms of constraint handling, as dual-mode predictions for the inputs are dynamic and converge asymptotically, then the number of constraints to be checked will be similar to that for the LMPC approach, that is $6NN_T$; this assumes that the dominant time constant is similar in both open-loop and closed-loop.

Conceptually, the constraint inequalities can be defined by a simple substitution of (27) into (7), for a sufficient horizon (say N_T). Following some simple algebra (Rossiter, 2018) the inequalities reduce to, for suitable $\mathcal{N}_d, \mathcal{M}_d$:

$$\mathcal{N}_d\mathbf{w}_k + \mathcal{M}_d\mathbf{x}_k \le \mathbf{f} \tag{28}$$

Performance index with DMPC Dual-mode predictions need more careful handling and it is generally considered that one should use infinite costing horizons in (2); indeed it is far more efficient to use infinite horizons rather than finite horizons because one can easily use Lyapunov equations to find the associated terms. Here we summarise the result only (Rossiter, 2018). A combination of input parameterisation (27) alongside system model (1), followed by substitution into performance index (2) results in:

$$J = \mathbf{w}_k^T S\mathbf{w}_k + \mathbf{w}_k^T P_c\mathbf{x}_k + \alpha_k; \tag{29}$$

for appropriate S, P_c. Significantly, the matrix S is block diagonal with each block the same, so the condition number of S does not change with the control horizon!

Remark 5. Early variants of DMPC (Scokaert and Rawlings, 1998) expressed the d.o.f. as the first n_u input values in (27) but it is generally accepted that the corresponding matrix S has worse conditioning than with the formulation given here (Rossiter et al., 1998).

4. NUMERICAL COMPARISONS

All the algorithms can be reduced to the following format. At each sample solve the following optimisation:

$$\min_{\mathbf{w}_k} J_k = \mathbf{w}_k^T S\mathbf{w}_k + \mathbf{w}_k^T P\mathbf{x}_k \ \text{ s.t. } \ \mathcal{N}\mathbf{w}_k + \mathcal{M}\mathbf{x}_k \le \mathbf{f} \tag{30}$$

The current value of input $\mathbf{u}_k$ is deduced from $\mathbf{w}_k$.

4.1 Monte Carlo and normalised approach to comparison

A number of randomised examples are generated with 2, 3 and 4 stable poles which are inside a circle of radius 0.95. The zeros are chosen randomly. In order to normalise the comparisons, the process steady-state gains are all unity, and consequently it is also reasonable to choose the weights in the performance index on the outputs and inputs to be unity. Given such a normalisation, one can assume relatively fixed input constraints such as:

$$|\Delta u_k| \le 0.5; \quad -1.3 \le u_k \le 1.3 \tag{31}$$

Changing these values will affect the comparison but this paper is looking for trends and generic messages which, inevitably, may not apply to specific examples. State constraints will be constrained to a scaled unit box; the scaling allowing some movement beyond a unit steady-state output. The output horizon n_y is taken to be large (circa 50) to ensure good practice guidance is met,

Table 1. Upper bound of number of constraint inequalities: N_T is 4 times the dominant time constant, N the system dimension and n_u the sample dimension of the d.o.f.

Algorithm	Inputs	Input rates	Outputs
GPC/DMC	$2Nn_u$	$2Nn_u$	$2NN_T$
Laguerre MPC)	$2NN_T$	$2NN_T$	$2NN_T$
Input Blocked MPC	$2Nn_u$	$2Nn_u$	$2NN_T$
Dual-mode MPC	$2NN_T$	$2NN_T$	$2NN_T$

although, given the random nature of the model generation and despite poles being faster than 0.95, even this is not large enough for all cases.

4.2 Constraint inequalities

A likely upper bound on the number of constraint inequalities required can be well approximated by the numbers in Table 1. It is pertinent to note that, apart from input constraints with GPC and simple blocking, the number of inequalities required is not expected to vary much with the choice of algorithm, largely because in order to ensure good confidence of recursive feasibility, state constraints need to be checked over a long horizon, irrespective of the horizons in the performance index and irrespective of n_u.

4.3 Unconstrained performance and dependence on n_u

In the unconstrained case, the DMPC algorithm is automatically optimal and therefore used as a benchmark for other algorithms. The optimal cost is:

$$J_{\text{runtime}} = \sum_{i=1}^{\infty} \|\mathbf{r} - \mathbf{y}_{k+i}\|_I^2 + \|\mathbf{u}_{k+i-1} - \mathbf{u}_{\text{ss}}\|_I^2 \qquad (32)$$

The percentage difference between (2) and (32) (i.e., performance loss against DMPC) of GPC, LMPC, and IBMPC algorithms are computed for $n_u \in [1, 9]$. We summarise the following observations:

- The use of the Laguerre input parameterisation may improve unconstrained performance with low n_u, but the benefit is by no means assured, may not be large/significant and can only be assessed on a case by case basis.
- The use of input blocking ameliorates unconstrained performance. The choice of appropriate block size is critical in reducing this effect.
- There are instances where all algorithms initially make the performance much worse at every n_u increment, which indicates the fundamental weakness of horizon tuning.

4.4 Conditioning of the QP optimisation

It has been shown (Rossiter et al., 1998) that for DMPC, using the closed-loop paradigm, the S matrix in (29) reduces to a diagonal where the diagonal blocks are identical. Consequently, for the SISO case, the condition number is one! Therefore, as in the previous section, DMPC can be used as a benchmark of having the best possible condition number. Figure 1 plots the logarithms of the condition numbers for GPC, LMPC, and IBMPC at $n_u = 9$ which clearly illustrates the computational aspects of deploying the three algorithms. We note the condition number for LMPC is consistently the worst and IBMPC the best. The implication is that LMPC would only be used wisely in cases where n_u is small. To our knowledge this observation has not appeared previously in the literature.

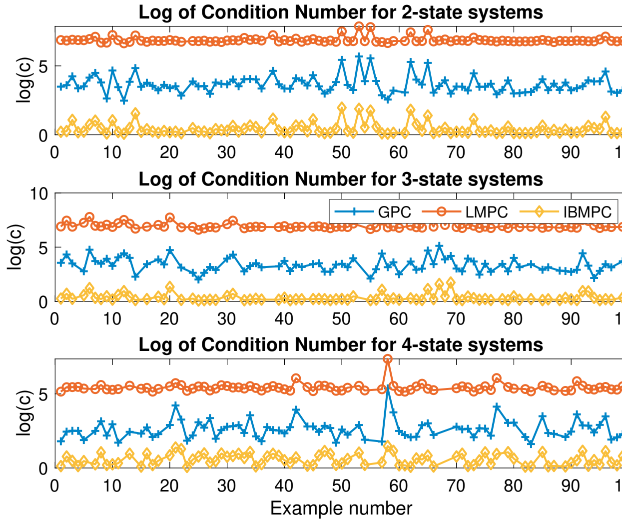

Fig. 1. Condition number of GPC, LMPC, and IBMPC for 2, 3, and 4-state examples with $n_u = 9$.

4.5 Feasible volumes comparison

Feasible volumes for DMPC is used as a benchmark as having relatively the smallest, and output constraints are added to ensure the feasible region is closed (bounded) and the volumes can be computed using the mpt3 toolbox (Herceg et al., 2013). Figure 2 plots the relative feasible volumes of the MAS ($n_u = 0$) and MCAS ($n_u = 3$) of GPC and LMPC algorithms to DMPC algorithm for 10 samples of 2, 3, and 4-state models. High relative volume for the MAS of some of the 2-state models indicates that DMPC algorithm yields very small MAS region. As reported in previous works (Rossiter et al., 2010), LMPC algorithm yields larger MCAS compared to GPC. However, for many example 4-state models, the computation of MCAS volumes for DMPC algorithm becomes unreliable and thus are not excluded.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an overview of four input parameterisation approaches in MPC, namely GPC/DMC, Laguerre MPC, Input-Blocked MPC, and Dual-Mode MPC. We set out a number of criteria to evaluate and compare these approaches. The different advantages and disadvantages of each approach are discussed based on sensible objective criteria. We compared the unconstrained closed-loop performance, the condition number, the number of constraint inequalities, and the relative feasible regions (MAS and MCAS) for small d.o.f. using numerical examples through Monte-Carlo simulations. Depending on priorities/example, the best approach changes.

REFERENCES

Abdullah, M. and Rossiter, J.A. (2021). Using Laguerre functions to improve the tuning and performance of

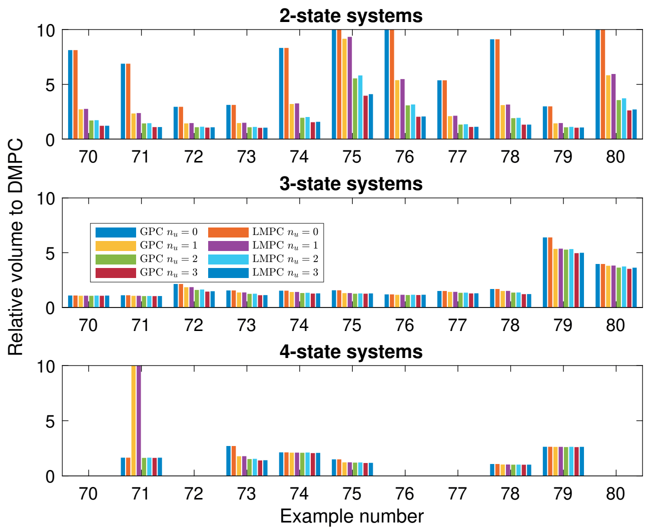

Fig. 2. Relative feasible volumes of GPC and LMPC to DMPC for 2,3, and 4-state examples.

predictive functional control. *International Journal of Control*, 94(1), 202–214.

Blanchini, F. and Miani, S. (2015). *Set-Theoretic Methods in Control*. Springer International Publishing, Cham.

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*. Society for Industrial and Applied Mathematics.

Cagienard, R., Grieder, P., Kerrigan, E.C., and Morari, M. (2007). Move blocking strategies in receding horizon control. *Journal of Process Control*, 17(6), 563–570.

Chisci, L., Rossiter, J.A., and Zappa, G. (2001). Systems with persistent disturbances: predictive control with restricted constraints. *Automatica*, 37(7), 1019–1028.

Clarke, D.W., Mohtadi, C., and Tuffs, P.S. (1987). Generalized Predictive Control Algorithm* Part I. The Basic. *Automatica*, 23(2), 137–148.

Clarke, D. and Scattolini, R. (1991). Constrained receding-horizon predictive control. *IEE Proc. D Control Theory and Applications*, 138(4), 347.

Cutler, C.R. and Ramaker, B.L. (1980). Dynamic matrix control – a computer control algorithm.

García, C.E., Prett, D.M., and Morari, M. (1989). Model predictive control: Theory and practice - A survey. *Automatica*, 25(3), 335–348.

Garone, E., Di Cairano, S., and Kolmanovsky, I. (2017). Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica*, 75, 306–328.

Gilbert, E.G. and Tan, K.T. (1991). Linear Systems with State and Control Constraints: The Theory and Application of Maximal Output Admissible Sets. *IEEE Trans. on Autom. Control*, 36(9), 1008–1020.

Grüne, L. and Pannek, J. (2017). Nonlinear Model Predictive Control. In *Nonlinear Model Predictive Control: Theory and Algorithms*, 45–69. Springer International Publishing, Cham.

Henson, M.A. (1998). Nonlinear model predictive control: current status and future directions. *Computers & Chemical Engineering*, 23(2), 187–202.

Herceg, M., Kvasnica, M., Jones, C.N., and Morari, M. (2013). Multi-Parametric Toolbox 3.0. 502–510.

Ishibuchi, H. (1995). MOGA: multi-objective genetic algorithms. In *IEEE International Conf. on Evolutionary Computation*.

Khan, B. and Rossiter, J.A. (2013). Alternative parameterisation within predictive control: a systematic selection. *International Journal of Control*, 86(8), 1397–1409.

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

Kouvaritakis, B., Rossiter, J., and Chang, A. (1992). Stable generalised predictive control: an algorithm with guaranteed stability. *IEE Proc. D Control Theory and Applications*, 139(4), 349.

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

Mayne, D.Q., Seron, M.M., and Raković, S.V. (2005). Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2), 219–224.

Pluymers, B., Rossiter, J.A., Suykens, J., and De Moor, B. (2005). The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty. In *Proc. of the 2005 American Control Conf.*, 804–809.

Qin, S.J. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764.

Rawlings, J.B. and Muske, K.R. (1993). The stability of constrained receding horizon control. *IEEE Trans. on Autom. Control*, 38(10), 1512–1516.

Rawlings, J.B. (2000). Tutorial Overview of Model Predictive Control. *IEEE Control Systems*, 20(3), 38–52.

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

Rossiter, J.A. (2006). A global approach to feasibility in linear MPC. *Proc. UKACC ICC*.

Rossiter, J.A. (2018). *A First Course in Predictive Control, Second Edition*. CRC Press, 2nd edition.

Rossiter, J.A. and Dughman, S.S. (2018). The impact of the input parameterisation on the feasibility of MPC and its parametric solution. *European Control Conf. (ECC)*, 300–305.

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

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

Valencia-Palomo, G. and Rossiter, J.A. (2012). Novel programmable logic controller implementation of a predictive controller based on Laguerre functions and multiparametric solutions. *IET Control Theory & Applications*, 6(8), 1003–1014.

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

Wang, L. (2009). *Model Predictive Control System Design and Implementation Using MATLAB*. Springer.