Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Exploiting Kautz functions to improve feasibility in MPC

B. Khan *, **J.A. Rossiter** * **and G. Valencia-Palomo** **

* *Automatic Control and Systems Eng., University of Sheffield, UK*
** *Instituto Tecnolgico de Hermosillo, Mexico*
e-mail: b.khan@sheffield.ac.uk., j.a.rossiter@sheffield.ac.uk

Abstract: This paper develops the recently published Laguerre MPC by proposing an alternative parametrization of the degrees of freedom in order to further increase the feasible region of model predictive control (MPC). Specifically, a simple but efficient algorithm that uses Kautz functions to parameterize the degrees of freedom in Optimal MPC is presented. It is shown that this modification gives mechanisms to achieve low computation burden with good feasibility and good performance. The improvements, with respect to an existing algorithm that uses a similar strategy, are demonstrated by examples.

Keywords: Predictive control, Feasibility, Kautz functions.

1. INTRODUCTION

Linear model predictive control (MPC) (Mayne et al., 2000; Rossiter, 2003; Camacho and Bordons, 2003) is well established and widely used both in industry and academia, but there are still some theoretical and practical issues which have non-satisfactory answers. For instance, one well understood conflict is between feasibility and performance. If a dual-mode MPC controller, that is one based on infinite output horizons, is tuned to give high performance, it will often have relatively small feasible regions (Scokaert and Rawlings, 1998; Kouvaritakis et al., 1998) unless one uses a prohibitively large number of decision variables (or degrees of freedom, d.o.f.). There is a pragmatic limit to increase the d.o.f. for the global feasible region as this compromises the computational burden. A strategy with the same number of d.o.f. giving good feasibility might be achieved through detuning of terminal mode but hence have relatively poor performance (Rossiter et al., 2010).

It has been noted that a DMC (Culter and Ramaker, 1980) or GPC (Clarke et al., 1987) type of algorithm will give reasonable performance for large (Large is typically 3-5 with sensible sampling rates) input horizons, so long as the output horizon is longer than the settling time. Significantly, DMC/GPC deploy a detuned terminal mode – essentially open-loop behavior. However, there are processes where this may not be so effective; for instance systems with: (i) poor open-loop dynamics and (ii) state or output constraints. In these cases, DMC or GPC with an input horizon of one may produce closed-loop behavior close to the open-loop and therefore unsatisfactory. State constraints may also severely restrict the operating region and have a strong influence on the constrained control law.

A further significant theoretical weakness of GPC/DMC is the lack of a general stability guarantee, especially during constraint handling. Although one could argue that with large output horizons such issues are nit picking, it can also be argued that if such guarantees are straightforward to achieve, then it seems reasonable to do so. Hence, in this paper the standard dual-mode prediction set up (Rossiter et al., 1998; Mayne et al., 2000) will be adopted as this enables guarantees of asymptotic stability and recursive constraint satisfaction.

Summarizing, typical trade-offs within linear MPC are: increase the feasible region, keep the computational load within sensible limits and obtain good closed-loop performance.

Several authors have worked on these trade-offs but the main research focus in recent years has moved towards nonlinear systems, robustness and parametric solutions. In stable systems with no state/output constraints, the simplest approach is to avoid constraint violations by using saturation (Rojas and Goodwin, 2002). Another simple approach (Tan and Gilbert, 1992) is defining multiple linear control laws and selecting online the current feasible law with the best performance. The weakness of this approach is that the optimal constrained control law is time-varying (Bemporad et al., 2002) and thus this can give suboptimal performance and restrictions to feasibility. Moreover, storing a multiple of control laws and in particular their corresponding feasible regions has a potentially large overhead, especially in the uncertain case (Pluymers et al., 2005).

Alternatively, interpolation techniques (Bacic et al., 2003; Rossiter et al., 2004) have utilized different formulations of the d.o.f. for optimization to enlarge the feasible region without too much detriment to performance. Moreover, another suggestion that has had little consideration in the literature is the concept of triple mode control. In this strategy one recognises that large feasible regions in conjunction with good performance often imply nonlinear or linear time varying (LTV) prediction dynamics (Rossiter et al., 2005; Imsland et al., 2008). The challenge is to find a suitable fixed[1] LTV control law which enlarges the feasible region without too much detriment to performance. This strategy provides good performance and feasibility at the expense of an increased offline computational burden and thus may be difficult for industrial implementation. More recently, Laguerre functions have been proposed as a means of parameterizing the input predictions in (Rossiter and Wang, 2008; Rossiter et al., 2010) as a simple way of improving the performance and feasibility. The main idea is to form the predictions as a combination of Laguerre polynomials. The use of Laguerre functions in predictive control have proven to be a very effective for improve feasibility and reducing the online computational burden.

This paper assumes that the terminal mode is well tuned and thus the only way to improve the feasibility is with the d.o.f. within or parameterisation of the predictions. Specifically the intent is to answer the question about Laguerre MPC, that is, why did you choose Laguerre polynomials and are these the only choice? Hence, in line with the proposals of (Rossiter et al., 2010), here Kautz functions are tested as these are more general than Laguerre functions. This paper will demonstrate that Kautz functions are an effective alternative to the standard basis set for parameterizing the d.o.f. within MPC and indeed may be more effective that Laguerre functions as they offer more variety in the key characteristics. The issue of making a systematic choice of 'function' or parameterisation of d.o.f. to best meet a specific objective is left as future work.

Section 2 will give the necessary background about modelling, predictive control and Laguerre optimal predictive control (LOMPC). Section 3 presents the basic properties of Kautz functions and compares them with Laguerre functions. Section 4 develops the novel Kautz OMPC (KOMPC) algorithm using Kautz functions to parameterize the input predictions. Section 5 gives numerical examples showing the efficacy of the proposed algorithm and this is followed by the conclusions in Section 6.

2. BACKGROUND

This section will introduce the background information and assumptions used in this paper.

2.1 Modelling and Optimal MPC

Assume discrete-time state-space model of the form:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \tag{1}$$

with $x_k \in \mathbb{R}^{n_x}$,$y_k \in \mathbb{R}^{n_y}$ and $u_k \in \mathbb{R}^{n_u}$ which are the state vectors, the measured output and the plant input respectively. Let the system be subject to constraints of the form

$$\begin{aligned} \underline{\Delta u} \leq \Delta u_k \leq \overline{\Delta u} \\ \underline{u} \leq u_k \leq \overline{u} \\ \underline{x} \leq x_k \leq \overline{x} \end{aligned} \tag{2}$$

The performance index (Scokaert and Rawlings, 1998) to be minimized with respect to $u_k, u_{k+1}...$ is

$$J = \sum_{i=0}^{\infty}(x_{k+i+1})^T Q(x_{k+i+1}) + (u_{k+i})^T R(u_{k+i})$$
$$s.t. \begin{cases} (1),(2) & \forall k \geq 0, \\ u_k = -\kappa x_k & \forall k \geq n_c \end{cases} \tag{3}$$

with Q and R positive definite state and input cost weighting matrices. Where κ is the optimal feedback gain minimizing 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 (Rossiter, 2003).

The prediction beyond n_c samples assumes only the optimal feedback, $u_k = -\kappa x_k$ is implemented (Imsland et al., 2008) and this is feasible only if the predicted state x_{n_c} is contained in a polytopic control invariant set (that is the MAS) for example:

$$\begin{aligned} &\mathcal{X}_{MAS} = \{x_0 \in \mathbb{R}^{n_x} \mid \underline{x} \leq x_k \leq \overline{x}, \\ &\underline{u} \leq -\kappa x_k \leq \overline{u}, x_{k+1} = Ax_k + Bu_k, \forall k \geq 0\} \end{aligned} \tag{4}$$

For simplicity of notation, the MAS can be described in the form $\mathcal{X}_{MAS} = \{x \mid Mx \leq d\}$ for suitable M and d. For convenience, the degrees of freedom with the future inputs are often reformulated in terms of a new variable c_k

$$\begin{aligned} u_{k+i} &= -\kappa x_{k+i} + c_{k+i}; & i &= 0, \ldots, n_c - 1 \\ u_{k+n_c+i} &= -\kappa x_{k+n_c+i}; & i &\geq 0 \end{aligned} \tag{5}$$

and hence the equivalent optimization to (3) is:

$$C^T SC \; s.t. \; Mx_k + NC \leq v; \; C = [c_k^T, \ldots, c_{k+n_c-1}^T]^T \tag{6}$$

The MCAS (maximal controlled admissible set) is defined as $\mathcal{X}_{MCAS} = \{x_k \in \mathbb{R}^{n_x} | \exists\, C \in \mathbb{R}^{n_c n_u}, Mx_k + NC \leq d\}$. Details of how to compute positive definite matrix S, matrices N, M and vector v are omitted as by now well known in the literature (Gilbert and Tan, 1991; Mayne et al., 2000; Rossiter, 2003). The optimal MPC (OMPC) algorithm is given by solving the QP optimization (6) at every sampling instant then implementing the first component of C, that is c_k in the control law of (5). When the unconstrained control law is not predicted to violate constraints (i.e. $x_k \in \mathcal{X}_{MAS}$), the optimizing C is zero so the control law is $u_k = -\kappa x_k$. The optimization of (6) can require a large n_c d.o.f. to obtain both good performance and a large feasible region.

2.2 LOMPC: Laguerre polynomials and MPC

Laguerre OMPC (LOMPC) is a dual-mode MPC algorithm (Rossiter and Wang, 2008; Rossiter et al., 2010) where the d.o.f. within the input predictions are parameterized in terms of Laguerre polynomials rather than using the more normal standard basis set. Laguerre polynomials are defined as follows:

$$L_i(z) = \sqrt{(1-p^2)}\frac{(z^{-1}-p)^{i-1}}{(1-pz^{-1})^i}; \qquad 0 \leq p < 1 \tag{7}$$

The input perturbations are reformulated as $c_{k+i} = L(i)^T\eta$ and hence the prediction cost becomes:

$$J = \eta^T[\sum_{i=0}^{\infty} A_L^i L(0) W L(0)^T (A_L^i)^T]\eta \tag{8}$$

where $L(i) = A_L L(i-1)$ (Details of how to define A_L are available in the references). Constraints representing the corresponding MCAS can also be summarised as: $\mathcal{X}_{MCAS} = \{x_k \in \mathbb{R}^{n_x} | \exists\, \eta \in \mathbb{R}^{n_c n_u}, Mx_k + NH_L\eta \leq d\}$.

[1] The time varying part refers solely to the prediction dynamics.

Algorithm 2.1: The LOMPC algorithm is summarized as:

$$\eta^* = arg\ min_\eta\ \ J_{LOMPC}\ \ s.t.\ \ Mx_k + NH_L\eta \ \le\ d \quad (9)$$

Define $C = L(0)^T\eta^*$ and implement the control law

$$u_k = -\kappa x_k + c_k$$

where c_k is the first element of the optimizing C.

3. KAUTZ FUNCTIONS

Laguerre functions (Rossiter et al., 2010; Wang, 2009) have been popular in filtering, system identification and control design because few parameters are enough to describe the behavior of the system. Its properties depend upon the selection of a scaling factor p and its complexity and accuracy of the description increases as the number of Laguerre networks increases. The application of Laguerre networks (Wang, 2009) is limited to a single pole selection (i.e. between 0-1). This limitation may be overcome by introducing Kautz networks which allow the selection of two complex poles which consequently may approximate system behavior better than using a single real pole as in Laguerre networks.

3.1 Kautz Network

Kautz networks were first proposed by Kautz (Kautz, 1954). The discrete-time Kautz network was generated from discretization of continuous-time Kautz network (a more detailed discussion on continuous time Kautz functions can be found in (Wang, 2009)).

The Kautz network is defined as follows

$$K_i(z) = \sqrt{(1-a^2)(1-b^2)}\frac{(z^{-1}-a)^{i-1}(z^{-1}-b)^{i-1}}{(1-az^{-1})^i(1-bz^{-1})^i}; \quad (10)$$

$$0 \le a < 1;\ \ 0 \le b < 1$$

where a and b are poles of the discrete-time Kautz network. The free parameters, a and b are selected by the user; these are also called the scaling factors. These functions are orthonormal and hence span the input prediction space effectively. However, the inverse z-transform of the Kautz networks do not lead to a compact expression of the Kautz functions in the time-domain so state-space representation is preferred and derived briefly here:

$$K_i(z) = K_{i-1}(z)\frac{(z^{-1}-a)(z^{-1}-b)}{(1-az^{-1})(1-bz^{-1})}; \quad (11)$$

$$0 \le a < 1;\ \ 0 \le b < 1$$

With $K_1(z) = \frac{\sqrt{(1-a^2)(1-b^2)}}{(1-az^{-1})(1-bz^{-1})}$. The discrete-time Kautz functions are expressed in a vector form as

$$K(n) = [k_1(n), k_2(n), \ldots, k_N(n)]^T \quad (12)$$

Taking advantage of the network realization in equation (11), the set of discrete-time Kautz functions satisfies the following difference equation

$$K(i+1) = GK(i) + FK(i-1) \quad (13)$$

where matrix G and F is $(N \times N)$ and is a function of parameters $\alpha = ab$, $\beta = (a+b)$ and $\gamma = \sqrt{(1-a^2)(1-b^2)}$, and, for example, in case of $N = 3$

$$G = \begin{bmatrix} \beta & 0 & 0 \\ \beta(\alpha-1) & \beta & 0 \\ \alpha\beta(\alpha-1) & \beta(\alpha-1) & \beta \end{bmatrix}; \quad K_{i-1}(0) = \gamma\begin{bmatrix} 1 \\ \alpha \\ \alpha^2 \end{bmatrix}$$

$$F = \begin{bmatrix} -\alpha & 0 & 0 \\ (1-\alpha^2) & -\alpha & 0 \\ \alpha(1-\alpha^2) & (1-\alpha^2) & -\alpha \end{bmatrix}; \quad K_{i-2}(0) = \gamma\beta\begin{bmatrix} 1 \\ \alpha-1 \\ \alpha(\alpha-1) \end{bmatrix} \quad (14)$$

In State-space form

$$\Psi(i+1) = \Sigma\Psi(i) + \Gamma\delta(i)$$
$$K(i) = \Phi\Psi(i) \quad (15)$$

where $\Psi(i) = [K_i\ K_{i-1}]^T$, $\Sigma = [G\ F;\ I\ 0]$, $\Gamma = [0\ 0]^T$ and $\Phi = [I\ 0]$. Conventional algorithms use d.o.f. (or perturbations c_k) signals that have an impact over just one sample and thus have a limited impact on feasibility. Essentially adding n_c or d.o.f. expands the feasible region to one where the state is able to enter the MAS with $C = 0$ in at most n_c steps; such an expansion may therefore be small in conventional algorithms. As it is normal to choose n_c to be small, say 1 or 2, then the volume of the feasible region is dominated by the choice of feedback κ in terminal mode.

Laguerre polynomials with $p > 0$ evolve over an infinite horizon and the speed of convergence is linked directly to the time constant p. If the best closed-loop input trajectory is expected to evolve with a given time constant, then it is intuitively obvious that an appropriate mix of Laguerre polynomials with this time constant is more likely to get close to the ideal trajectory than a mix of simple input values over a short infinite horizon. In a similar vein, where a state is a long way from the unconstrained feasible region, a small number of simple input perturbations is not sufficient to regain feasibility and hence conventional algorithms may have poor feasibility. Laguerre functions (Rossiter and Wang, 2008; Rossiter et al., 2010) provide an alternative trajectory for improving the feasibility with the same number of d.o.f..

A Kautz function with $a = p$, $b = p$ where p is real, gives:

$$K(i) = L(i).$$

which demonstrates that Laguerre functions are a special case of Kautz functions. clearly therefore, Kautz functions have more flexibility than Laguerre functions and it is this flexibility that is explored in this paper.

For completeness and to improve insight, Fig. 1, shows the coefficients of the first four Kautz (with poles at $0.8 \pm 0.4j$ and Laguerre polynomials ($p = 0.8$). In both cases the speed of convergence is linked with the poles. In Fig. 1 the convergence of Kautz polynomials are slower than that of Laguerre polynomials due to the poles being closer to the unit circle, but the oscillatory behaviour still allows the capture of some faster dynamics; hence we expect Kautz functions to offer more functionality than Laguerre polynomials by both facilitating slower convergence (to improve feasibility) in conjunction with more rapid transitions if required in near transients to improve performance.

4. USING KAUTZ POLYNOMIALS IN OMPC

A fundamental weakness of the OMPC algorithm is infeasibility when n_c steps are insufficient to move the initial

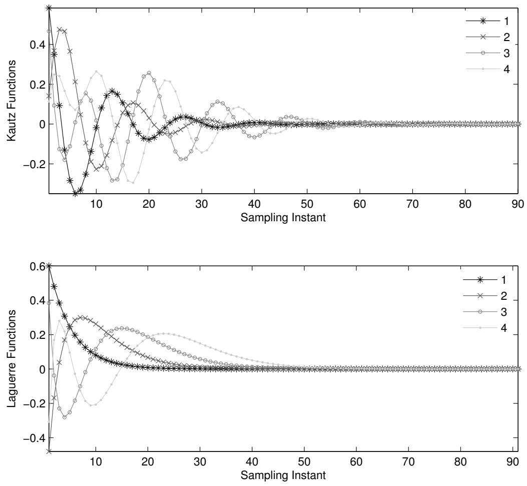

Fig. 1. Coefficient of four Kautz and Laguerre polynomials

state into the MAS. This weakness can be overcome by increasing the d.o.f. to allow more steps for reaching the MAS, but obviously at the expense of an increased computational burden. Another way of increasing the feasible region is by detuning the terminal mode which may compromise performance. However, an alternative hightlighted in Rossiter et al. (2010) is to parameterise the d.o.f. differently so that the impact on the input predictions is over a longer horizon, thus relaxing the time requirement for entering the MAS. This section derives an algorithm which uses Kautz functions for this parameterisation, whereas the next section will compare these with the earlier Laguerre based approach.

4.1 Kautz functions in OMPC

There are two complex poles a and b that define the time scale for the input predictions using a combination of Kautz functions and one could even consider modifying a and b online to improve feasibility where required and thus deploying no extra d.o.f.. The algorithm associated using Kautz functions is denoted as KOMPC for Kautz OMPC. Kautz functions can easily be used to redesign DMC/GPC achieving good performance and feasibility, but here focus is on dual-mode algorithms with guaranteed stability.

4.2 Kautz OMPC or KOMPC

Kautz functions are used to parameterize the perturbations c_k around the unconstrained optimal. The prediction using decision variables used in OMPC and KOMPC are put side by side (ρ denotes the KOMPC d.o.f.):

$$C = \underbrace{\begin{pmatrix} c_k \\ \vdots \\ c_{k+n_c-1} \\ 0 \\ 0 \\ \vdots \end{pmatrix}}_{\text{OMPC}} \; or \; C = \begin{pmatrix} K(0)^T \\ K(1)^T \\ \vdots \\ \vdots \\ \vdots \end{pmatrix} \rho = \underbrace{H_K \rho}_{\text{KOMPC}} \qquad (16)$$

The key difference here from OMPC is that the H_K matrix has large number (in fact infinite) of rows. The performance index J can be computed in terms of perturbation c_k as:

$$J = \sum_{i=0}^{\infty} c_{k+i}^T W c_{k+i} \qquad (17)$$

However, from eqn.(16) note that $c_{k+i} = K(i)^T \rho$ and from eqn.(13) the new performance index becomes:

$$\begin{aligned} J_{KOMPC} &= \sum_{i=0}^{\infty} \rho^T K(i) W K(i)^T \rho \\ &= \rho^T \; [\sum_{i=0}^{\infty} K_1(0)^T (G^i)^T W G^i K_1(0)] \; \rho + \\ & \rho^T [\sum_{i=0}^{\infty} K_0(0)^T (F^i)^T W F^i K_0(0)] \; \rho \end{aligned} \qquad (18)$$

The Feasible region can also be rewritten in the form: $\mathcal{X}_{MCAS} = \{x_k \in \mathbb{R}^{n_x} | \exists \rho \in \mathbb{R}^{n_c n_u}, Mx_k + NH_K \; \rho \; \leq \; d\}$.

Algorithm 4.1: The KOMPC algorithm is summarized as:

$$\rho^* = arg \; min_\rho \;\; J_{KOMPC} \;\; s.t. \; Mx_k + NH_K \; \rho \; \leq \; d \quad (19)$$

Define $C = K(0)^T \rho^*$ and implement the control law

$$u_k = -\kappa x_k + c_k$$

where c_k is the first element of the optimizing C.

5. NUMERICAL EXAMPLES

This section will illustrate the efficacy of the proposed KOMPC algorithm in comparison with LOMPC and OMPC using numerical examples. The main focus of this paper is a comparison based on performance and feasibility. The results will be presented in a way that is suitable for any number of state dimensions. The closed loop performance is measured by computing the performance index J over the time span where system converges. The optimal performance index J_{opt} is computed using OMPC with high n_c ($n_c = 20$ is used for numerical examples). The plots show the normalized performance index for comparison. The feasible regions in general are difficult to compare visually when larger than 2D plots. The volume or extent of the feasible region is computed by selecting different state directions and computing, relatively, how far out in these directions a feasible solution exists. The maximum distance point for various direction is denoted by σ, the various algorithms are then tested for initial points $\lambda\sigma$ ($0 \leq \lambda \leq 1$). Clearly the larger the λ for which they are feasible, the larger the feasible region in that specific direction. Infeasibility is denoted by a zero in the normalized performance index plots.

5.1 ***Example 1***

The discrete-time state-space model and constraints are

$$A = \begin{bmatrix} 0.6 & -0.4 \\ 1 & 1.4 \end{bmatrix}; \; B = \begin{bmatrix} 0.2 \\ 0.05 \end{bmatrix}; \quad C = [1 \;\; -2.2]; \qquad (20)$$

$$\overline{\Delta u} = 0.4 = -\underline{\Delta u}; \; \overline{u} = 0.8; \; \underline{u} = -1.5; \; \overline{y} = 5 = -\underline{y}$$

The tuning parameters are $Q = I_{2\times2}$,$R = 2$, $n_c = 2$, $p = 0.8$, $a = 0.8 + 0.45j$ and $b = 0.8 - 0.45j$.

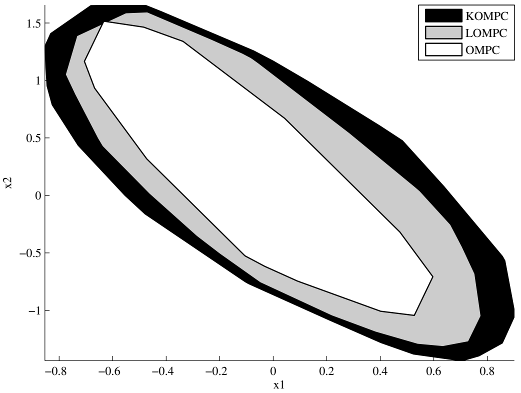

Fig. 2. Comparison of MCAS for $n_c = n_\rho = 2$ for KOMPC,LOMPC,OMPC algorithms

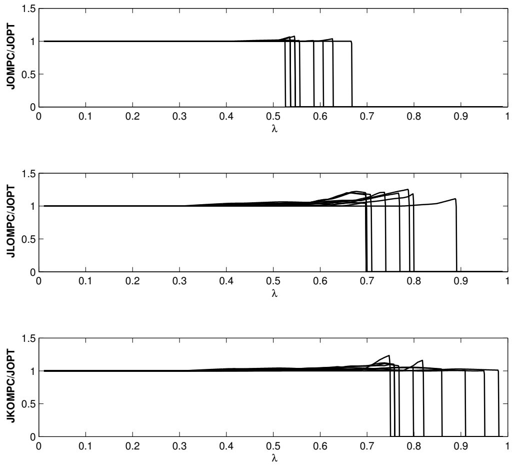

Fig. 3. Normalised performance index JOMPC/JOPT,JLOMC/JOPT,JKOMPC/JOPT for various state directions.

Figure 2 shows the feasible regions from which it is clear that KOMPC has a larger MCAS than both LOMPC and OMPC for the same number of d.o.f. i.e. $n_c = 2$. The plots of normalized cost against λ for a number of different state directions are plotted in Fig. 3 for OMPC, LOMPC and KOMPC. The global optimal is computed with OMPC with $n_c = 20$.

- OMPC gives good performance for states well within the MCAS, but feasibility is severely restricted as the plots drops to zeros for small λ.
- LOMPC has better feasibility than OMPC but loses some performanc compared to the global optimal near its own MCAS boundary.
- KOMPC gives both better feasibility and performance that both OMPC and LOMPC.

5.2 *Example 2*

For this example the discrete-time state-space model and constraints are given by

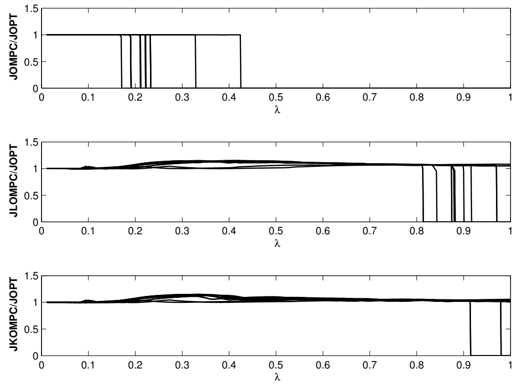

Fig. 4. Normalised performance index JOMPC/JOPT,JLOMC/JOPT,JKOMPC/JOPT for various state directions.

$$A = \begin{bmatrix} 1.4000 & -0.1050 & -0.1080 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}; \; B = \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix};$$

$$C = [5 \;\; 7.5 \;\; 0.5]; \qquad (21)$$

$$\overline{\Delta u} = 0.02 = -\underline{\Delta u}; \; \overline{u} = 0.4 = -\underline{u}; \; \overline{y} = 1.2 = -\underline{y}$$

The tuning parameters are $Q = C^TC$,$R = 2$, $n_c = 2$, $p = 0.8$, $a = 0.8 + 0.26j$, $b = 0.8 - 0.26j$ and ten state directions are chosen for the initial states.

Fig. 4 shows the feasibility/performance results for OMPC, LOMPC and KOMPC algorithms.

- OMPC has good performance while feasible, but very limited feasibility.
- LOMPC has noticeably better feasibility than OMPC, but with a small loss of performance in mid ranges of λ.
- KOMPC has better feasibility than both OMPC and LOMPC and less performance loss than LOMPC.

5.3 *Example 3*

For this example the discrete-time state-space model and constrained are

$$A = \begin{bmatrix} 0.9146 & 0 & 0.0405 & 0.1 \\ 0.1665 & 0.1353 & 0.0058 & -0.2 \\ 0 & 0 & 0.1353 & 0.5 \\ -0.2 & 0 & 0 & 0.8 \end{bmatrix}; \quad B = \begin{bmatrix} 0.0544 & -0.0757 \\ 0.0053 & 0.1477 \\ 0.8647 & 0 \\ 0.5 & 0.2 \end{bmatrix};$$

$$C = \begin{bmatrix} 1.7993 & 13.2160 & 0 & 0.1 \\ 0.8233 & 0 & 0 & -0.3 \end{bmatrix}. \qquad (22)$$

$$\overline{\Delta u} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = -\underline{\Delta u}; \; \overline{u} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} = -\underline{u}; \; \overline{y} = \begin{bmatrix} 10 \\ 10 \end{bmatrix} = -\underline{y}$$

The tuning parameters are $Q = I_{4\times4}$,$R = I_{2\times2}$, $n_c = 2$, $p = 0.8$, $a = 0.8 + 0.45j$,$b = 0.8 - 0.45j$ and fifteen state directions are chosen for the initial states. Fig. 5 shows the comparison results of performance/feasibility for OMPC, LOMPC and KOMPC algorithms for chosen state directions. In this case there is a little to choose between the algorithms which demonstrates that a generic result is not possible - using Kautz is a tool which sometimes has major benefits and other times may not be required.

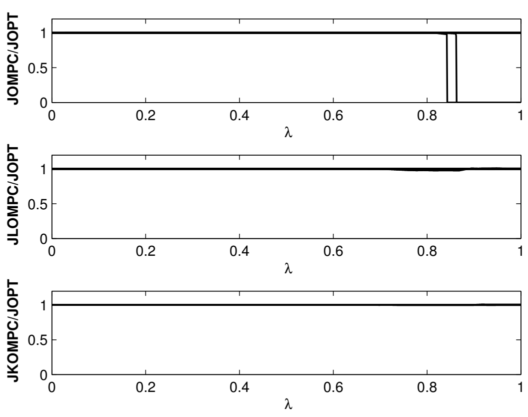

Fig. 5. Normalised performance index JOMPC/JOPT,JLOMC/JOPT,JKOMPC/JOPT for various state directions.

Table 1. Volume comparison

	OMPC	LOMPC	KOMPC
Example 1	1.3841	2.2723	2.779
Example 2	0.0035	0.0163	0.0179
Example 3	71.6438	73.1401	73.3729

For completeness, Table 1 shows the volumes of the MCAS for the three examples presented in this section. This data is an objective measure of feasibility and the observation that KOMPC may improve the feasibility by utilizing the d.o.f. more effectively.

6. CONCLUSION

The paper has argued for the potential benefits of Kautz functions as an alternative parameterization for maximizing the feasible region in conventional MPC algorithms with a fixed number of d.o.f.. It has been shown through examples that feasibility can be improved without degrading the performance. It has also been shown that a simple re-parameterization of the degrees of freedom within the input predictions can achieve better performance and feasibility. However, of more significance, the paper has tackled the question concerning the earlier proposed use of Laguerre functions to parameterize the d.o.f. in the predictions and clearly demonstrated that obvious alternatives do exist and in fact, this paper indicates that Kautz functions may indeed be preferable to Laguerre in general.

The authors believe that the field of how to increase the feasible region, improve the performance and lower the computational burden using alternative parameterizations within the predictions clearly merits more work. This paper has focussed on just one possible parameterisation and future work will tackle the question of whether there exists a 'systematic' method for choosing the best parameterisation for any given problem. Moreover, it has been noted that computational burden is linked not only to the number of d.o.f. but also the optimisation structure and thus the next step is to consider how easily one can tailor a quadratic programming problem to KOMPC and thus gain further computational benefits.

REFERENCES

Bacic, M., Cannon, M., Lee, Y., and Kouvaritakis, B. (2003). General interpolation in MPC and its advantages. *IEEE Trans. AC*, 48(6), 1092–1096.

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

Camacho, E. and Bordons, C. (2003). *Model predictive control.* Springer, London.

Clarke, D., Mohtadi, C., and Tuffs, P. (1987). Generalised predictive control – Part 2. Extension and interpretations. *Automatica*, 23(2), 137–160.

Culter, C. and Ramaker, B. (1980). Dynamic matrix control – A computer control algorithm. *Proc. ACC.*

Gilbert, E. and Tan, K. (1991). Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Trans. on Automatic Control*, 36(9), 1008–1020.

Imsland, L., Rossiter, J., Pluymers, B., and Suykens, J. (2008). Robust triple mode MPC. *IJC*, 81(4), 679–689.

Kautz, W. (1954). Transient synthesis in the time domain. *IEEE Trans. IRE Transections on Circuit Theory*, 1(3), 29–39.

Kouvaritakis, B., Rossiter, J., and Cannon, M. (1998). Linear quadratic feasible predictive control. *Automatica*, 34(12), 1583–1592.

Mayne, D., Rawlings, J., Rao, C., and Scokaret, P. (2000). Constrained model predictive control. *Automatica*, 36(6), 789–814.

Pluymers, B., Rossiter, J., Suykens, J., and DeMoor, D. (2005). The efficient computation of polyhedral invarient sets for linear sysetms with polytopic incertainty. *Proc. ACC.*

Rojas, O. and Goodwin, G. (2002). A simple antiwindup strategy for state constrained linear control. *IFAC World Congress.*

Rossiter, J. (2003). *Model-based predictive control, a practical approach.* CRC Press, London.

Rossiter, J., Kouvaritakis, B., and Bacic, M. (2004). Interpolation based computationally efficient predictive control. *IJC*, 77(3), 290–301.

Rossiter, J., Kouvaritakis, B., and Cannon, M. (2005). An algorithm for reducing complexity in parametric predictive control. *IJC*, 78(18), 1511–1520.

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

Rossiter, J. and Wang, L. (2008). Exploiting Laguerre functions to improve the feasibility/performance compromise in MPC. *Proc. CDC.*

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

Scokaert, P. and Rawlings, J. (1998). Constrained linear quadratic regulation. *IEEE Trans. on Automatic Control*, 43(8), 1163–1168.

Tan, K. and Gilbert, E. (1992). Multimode controllers for linear discrete time systems with general state and control constraints. *World Scientific.*

Wang, L. (2009). *Model Predictive Control System Design and Implementation Using MATLAB.* Springer, 3 Apple Hill Drive, Natick,MA 01760-2098, USA.