Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Exploiting Laguerre functions to improve the feasibility/performance compromise in MPC

J.A. Rossiter* and Liuping Wang**

Abstract— **This paper develops a novel mechanism for managing the performance and feasibility trade off within predictive control. Specifically, it demonstrates that the potential of Laguerre functions needs more investigation by the community as this offers a simple and effective alternative to the more usual parameterisations using pulse operator in the predictions. The paper develops a simple and efficient dual-mode algorithm using Laguerre functions and demonstrates, by example, the improvements with respect to a conventional approach, in terms of performance and feasibility benefits.**

I. Introduction

Although linear predictive control [6], [10], [2] is well established and widely used, there are still some theoretical and practical issues which have non-satisfactory answers. One well understood conflict is that between feasibility and performance. A controller that is well tuned to give high performance will often have relatively small feasible regions [14], [5] unless one uses a prohibitive number of decision variables (or degrees of freedom, d.o.f.). Conversely, a strategy giving good feasibility may will achieve this through de-tuning and thus have relatively poor performance.

Several authors have considered this problem from different angles. The simplest approach is to adopt saturation control when the controller is infeasible [9], but this is only really applicable to stable systems with no output/state constraints and, may give poor performance where the open loop dynamics are poor. A similar but slightly more complex solution was proposed in [15] where the strategy is to use a look up table and choose from a set of predetermined controllers, the best performing of the feasible options. Both these strategies do not consider in detail the observation made most clear in parametric methods [1], that is the constrained optimal sequence has affine dependence on the state and is linear time varying; thus it seems logical to pursue this latter avenue.

Some interesting work considered so called *triple mode* strategies [12], [4] where one embeds a smooth transition between a controller with good feasibility and one with good performance into a single model and use the decision variables to improve performance/feasibility further still. This work is successful but relies on heavy computation and algebra in the off-line parts and thus may be difficult for industrialists to implement.

*Department of Automatic Systems and Control Engineering, Mappin Street, University of Sheffield, S1 3JD, UK. j.a.rossiter@sheffield.ac.uk

**School of Electrical and Computer Engineering, RMIT, Australia. liuping.wang@rmit.edu.au

This paper considers another alternative which is strongly related to a theme the first author has pursued for several years. It seems reasonable to structure the input predictions such that the desired behaviour is included in the class of possible optimisation outcomes; this way the optimisation is able to give an LQR optimal (e.g. [14], [13]). Such a structure implies that the predictions evolve over an infinite horizon, as opposed to the finite horizon input predictions common in DMC and GPC. The difficulty is that using the LQR optimal as the main base can give very poor feasibility [5] with low numbers of d.o.f. and this because the decision variables are commonly taken as perturbations to the optimal trajectory at just a few, say n_c, points; hence the initial state is restricted to the points that can reach a small unconstrained feasible region in just n_c samples. One might overcome this limitation if the class of predictions included other trajectories that also evolve over an infinite horizon, thus relaxing the time scale required to enter a target set. Interpolation methods [11] are one way of augmenting the prediction set while maintaining a tight hold on the number of d.o.f., but this paper aims to consider a relatively novel and under publicised alternative which has strong links to the PFC algorithm [8].

Perhaps going against the common trend of the last decade, the proposal is to make use of open loop predictions. This may seem opposed to the more usual terminal (or dual) mode approaches but as the reader will see, elements of both are required and this is one aspect of the novelty. Conceptually, the main thrust of this paper is to interpolate between predictions which evolve over a long horizon rather than the more usual finite horizon. One argument [16], [17] is that the typical choices of d.o.f. in GPC and DMC were chosen for convenience and transparency, not because they were the best way to map the desired set. There exist other othorgonal choices of trajectory such as Laguerre functions which may introduce other benefits. A parallel but related argument made by proponents of PFC is that one should know roughly what the desired closed-loop dynamic is, and ensure this is explicitly included in the design process.

In summary this paper will make two main contributions. First it will revisit the use of Laguerre functions with a standard DMC/GPC type of algorithm and demonstrate clearly the potential benefits. Secondly, these insights will be transferred to a more sophisticated dual mode algorithm and it will be shown that, in some cases, one can obtain huge feasibility improvements over a standard algorithm such as [14] without necessarily any sacrifice of performance. We should point out that, in our view, theoretical proofs of which method is better are unlikely to be tractable and useful and

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

it is more important to add the proposed approach to the toolkit so that it is available when it gives clear benefits. Section II will give the necessary mathematical background on conventional MPC algorithms and Laguerre functions, then section III will develop novel MPC algorithms using both open-loop and closed-loop predictions but combining Laguerre functions with the decision variables. Section IV gives numerical examples of the efficacy of the proposed algorithms and is followed by conclusions.

II. Modelling, predictive control and Laguerre

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

A. Model, constraints and integral action

Assume a standard state space model of the form:

$$x_{k+1} = Ax_k + Bu_k; \quad w_k = Cx_k + d_k \tag{1}$$

with d_k the disturbance; let the measured plant output be given as y_k. We will use an independent model approach to prediction and disturbance estimation, so if w is the output of the independent model (given by simulating model (1) in parallel with the plant), then the disturbance estimate is $\hat{d}_k = y_k - w_k$.

Disturbance rejection and offset free tracking will be achieved using the offset form of state feedback [7], that is:

$$u_k - u_{ss} = -K(x_k - x_{ss}) \tag{2}$$

where x_k is the state of the independent model and x_{ss}, u_{ss} are estimated values of the steady-states giving no offset; these depend upon the model parameters, the set point r and the disturbance estimate. For simplicity (for details M_1, M_2 see [7]) define:

$$x_{ss} = M_1(r_k - \hat{d}_k); \quad u_{ss} = M_2(r_k - \hat{d}_k) \tag{3}$$

Let the system be subject to constraints of the form

$$\left.\begin{array}{l} \underline{u} \le u_k \le \overline{u} \\ \underline{\Delta u} \le \Delta u_k \le \overline{\Delta u} \\ \underline{y} \le y_k \le \overline{y} \end{array}\right\} \quad \forall k \tag{4}$$

B. Predictive control - DMC/GPC

In the literature, DMC/GPC algorithms are derived based on control increments (e.g. [10]). However, to ensure consistency with the dual-mode approaches discussed later, the cost function and predictions are based on offset from steady-state rather than control increments.

Open-loop output predictions over some horizon n_y can be computed to have the following form[1]:

$$\underrightarrow{y}_k = H\,\underrightarrow{u}_{k-1} + Px_k + L\hat{d}_k \tag{5}$$

[1]The state x_k is obtained from the independent model.

where the matrices H, P depend on the model parameters and horizons, L is $[I, I, ..., I]^T$ and the arrow notation (right for future) is defined as follows:

$$\underrightarrow{y}_k = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+n_y} \end{bmatrix}; \quad \underrightarrow{u}_{k-1} = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+n_u-1} \end{bmatrix} \tag{6}$$

Typically for DMC/GPC it is assumed that $u_{k+n_u+i} = u_{ss},\ i \ge 0$ with $n_u << n_y$. The performance index is given as:

$$J = \|W_y(Lr_k - \underrightarrow{y}_k)\|_2^2 + \|W_u(\underrightarrow{u}_{k-1} - Lu_{ss})\|_2^2 \tag{7}$$

where $W_y,\ W_u$ are tuning weights. The control law is given by the first component of:

$$\underrightarrow{u}_{k-1} = \arg\min_{\underrightarrow{u}_{k-1}} \quad J \ \text{ s.t. } (4) \tag{8}$$

C. Optimal MPC or OMPC

In the case of optimal MPC [14], [13], the predictions are based around the unconstrained optimal (essentially what you would get with the same cost function based on infinite input and output horizons), hence define the unconstrained optimal feedback minimising J for infinite horizons to be K_o. By embedding this optimal into the prediction class, then optimal behaviour should ensue whenever that is feasible (satisfies constraints).

To allow for constraints it is necessary to allow perturbations about the unconstrained optimal predictions, hence define the perturbed input predictions at the current sample as:

$$u_{k+i} - u_{ss} = -K_o(x_{k+i} - x_{ss}) + c_{k+i} \tag{9}$$

with $c_{k+i} = 0, i \ge n_c$. It can be shown easily [10] that substituting this prediction into the cost J with infinite horizons, the cost takes the form:

$$J = \sum_{i=0}^{n_c-1} c_{k+i}^T S c_{k+i} \tag{10}$$

with $S = B^T\Sigma B + R,\ \Sigma - \Phi^T\Sigma\Phi = Q + K_o^T R K_o,\ \Phi = A - BK_o$. Hence the MPC algorithm, here denoted as OMPC for optimal MPC, is given from:

$$\underrightarrow{c}_{k-1} = \arg\min_{\underrightarrow{c}_{k-1}} \quad \sum_{i=0}^{n_c-1} c_{k+i}^T S c_{k+i} \ \text{ s.t. } (4) \tag{11}$$

The first component of $\underrightarrow{c}_{k-1}$ is used to define u_k from (9).

D. Laguerre polynomials

Laguerre polynomials are defined as follows:

$$L_i(z) = \sqrt{(}1 - a^2)\frac{(z^{-1} - a)^{i-1}}{(1 - az^{-1})^i}; \quad 0 \le a < 1 \tag{12}$$

These are othornormal and hence span the input prediction space effectively.

The key point to note is that conventional algorithms such as GPC/OMPC summarised in the earlier sections use d.o.f.

over just the next n_u, n_c samples because these assume in prediction that $u_{k+i} = u_{ss}, i \geq n_u$ or $c_{k+i} = 0, i \geq n_c$. Usually $n_u,\ n_c$ are chosen to be small, say one or two, and hence there is little flexibility in the prediction class.

Laguerre polynomials with $a > 0$ evolve over an infinite horizon and the speed of convergence is linked directly to the 'time scaling factor' a. 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 scaling factor is more likely to get close to the ideal trajectory than a mix of simple input values (i.e. equivalent to using pulse operators) over a short finite 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 [5] and hence conventional algorithms may have poor feasibility. Laguerre provides a more complete alternative trajectory and thus may improve feasibility.

Remark 2.1: One can reconstitute the normal GPC decision variables (i.e. $u_k, u_{k+1}, ...$) by noting that these are replicated by Laguerre with $a = 0$. Hence, if $a = 0$ then $L_0 = [1, 0, 0, \ldots],\ L_1 = [0, 1, 0, 0, \ldots],$ etc.

For completeness, a means of computing the sequences for any a is given in appendix 1. Specifically this shows how one might build up an input prediction $\underset{\rightarrow}{u}_{k-1}$ from an appropriate mix of Laguerre polynomials.

E. Summary

This section has summarised two popular MPC algorithms in their usual form and argued that replacing the more usual decision variables of $u_{k+i},\ c_{k+i}$ with Laguerre polynomials L_i may bring about benefits. The remainder of this paper shows how this might be done and demonstrates the potential benefits of doing so.

III. Using Laguerre polynomials in GPC and OMPC

This section derives GPC/DMC and OMPC algorithms which use a mix of Laguerre polynomials as the decision variables. For ease of distinction these are denoted as LDMC and LOMPC for Laguerre DMC and Laguerre OMPC respectively. As the algebra is relatively routine, the presentation is deliberately concise.

A. A Laguerre DMC algorithm

The DMC form of the prediction was given in (5). The only change being proposed here is to replace the decision variables in the future control sequence. The reader is reminded that it is important with the disturbance model adopted to include the estimated steady-state value of the input explicitly and optimise around this, hence we use the Laguerre polynomial as a perturbation about the expected steady-state.

To better enable the reader to see the differences, here we place the DMC and LDMC input prediction sequences side by side:

$$\underset{\rightarrow}{u}_{k-1} = \underbrace{\begin{bmatrix} u_k \\ \vdots \\ u_{k+n_u-1} \\ u_{ss} \\ u_{ss} \\ \vdots \end{bmatrix}}_{DMC} \quad or \quad \underset{\rightarrow}{u}_{k-1} = H_L\eta + \underbrace{\begin{bmatrix} u_{ss} \\ \vdots \\ u_{ss} \\ u_{ss} \\ u_{ss} \\ \vdots \end{bmatrix}}_{LDMC} \tag{13}$$

where H_L is defined in appendix 1. The LDMC output predictions are determined by substituting the input predictions into (5), that is:

$$\underset{\rightarrow}{y}_k = H[H_L\eta + Lu_{ss}] + Px_k + L\hat{d}_k \tag{14}$$

Here after everything is standard except that the constraints are parameterized using the same Laguerre parameter vector η (see [17]). Substitute predictions (14) into the cost function ((7) such that:

$$\begin{aligned} J_{LDMC} &= \|W_y(Lr_k - H[H_L\eta + Lu_{ss}] + Px_k + L\hat{d}_k)\|_2^2 \\ &+ \|W_uH_L\eta\|_2^2 \end{aligned} \tag{15}$$

Next minimise J_{LDMC} wrt η and subject to constraints

$$\eta = \arg\min_{\eta} \quad J_{LDMC} \ \text{ s.t. } (4) \tag{16}$$

The control law is given by substituting the optimum η into (13b) and then implementing the first value of the predicted input trajectory, that is by taking the top row of (13b).

B. Laguerre OMPC or LOMPC

OMPC is, by design, based around the unconstrained optimal sequence. This basic concept is preserved and thus the Laguerre polynomials are used, not to model the input trajectories directly but rather the perturbations c_k around the unconstrained optimal (which already includes u_{ss}). For ease of viewing and in a similar fashion to the previous section, the corresponding predictions for the decision variables used in OMPC and LOMPC are put side by:

$$\underset{\rightarrow}{c}_{k-1} = \underbrace{\begin{bmatrix} c_k \\ \vdots \\ c_{k+n_c-1} \\ 0 \\ 0 \\ \vdots \end{bmatrix}}_{OMPC} \quad or \quad \underset{\rightarrow}{c}_{k-1} = \underbrace{H_L\eta}_{LOMPC} \tag{17}$$

The reader will note that one key difference here is that the H_L matrix has a large number of rows[2] and, as mentioned before, sufficient rows must be included to capture all the dynamics within the constraint handling. To ensure a a proper synergy with the OMPC algorithm and to allow strict statements about recursive feasibility and convergence, we

[2]Technically infinite but as with all MPC it usual to go up a given horizon only or capture the asymptotic behaviour with Lyapunov equations.

will demonstrate how easily the corresponding cost J can be computed which includes the entire implied dynamic.

1) For LOMPC the predicted cost can be represented in terms of perturbations c_k as:

$$J = \sum_{i=0}^{\infty} c_{k+i}^T S c_{k+i} \tag{18}$$

2) Next, note that for LOMPC we define $c_{k+i} = L(i)^T \eta$ where $L(i)$ is a multivariable equivalent if required (see appendix 1).
3) Finally, substitute into (18) to give

$$J_{LOMPC} = \sum_{i=0}^{\infty} \eta^T L(i) S L(i)^T \eta \tag{19}$$

4) Next, substitute from equation (22) which has $L(i) = A_L L(i-1)$ and hence:

$$J_{LOMPC} = \eta^T \underbrace{[\sum_{i=0}^{\infty} A_L^i L(0) S L(0)^T (A_L^i)^T]}_{S_L} \eta \tag{20}$$

Hereafter everything is standard. Find the η which minimises cost (20) wrt η subject to the corresponding predictions meeting constraints.

$$\eta = \arg\min_{\eta} \quad J_{LOMPC} \quad \text{s.t.} \quad (4) \tag{21}$$

Then reconstitute the first value of the predicted input trajectory u_k using $c_k = L(0)^T \eta$ and (9).

Remark 3.1: It is straightforward to show, with conventional arguments, that the LOMPC algorithm gives guaranteed stability in the nominal case and offset free tracking when that is feasible.

IV. Numerical Examples

This section will illustrate the efficacy of the proposed LDMC, LOMPC algorithms by way of numerical examples.

A. Explanation of illustrations or comparisons

Specifically the aim here is to compare two aspects:

- The closed-loop performance for a range of initial conditions.
  This is measured by computing the performance index J over a time span long enough for the system to have converged. The global optimum (the OMPC algorithm with high n_c) is used a measure of how far the algorithms are from optimal. The figures typically show the ratio of a given algorithm cost to the global optimal cost so the nearer to one the better; however a zero denotes infeasibility.
- The volume or extent of the feasible region for a range of state directions.
  As it is difficult to plot in higher than 2D space, instead we choose state directions and compute how far out in these directions a feasible solution exists; the maximum distance points for various directions could be denoted p_i. The various algorithms are then tested for initial points $\lambda p_i,\ 0 \le \lambda \le 1$. Clearly the larger lambda for which they are feasible, the larger the feasible region in that specific direction. Infeasibility is denoted by a zero in the cost plots.

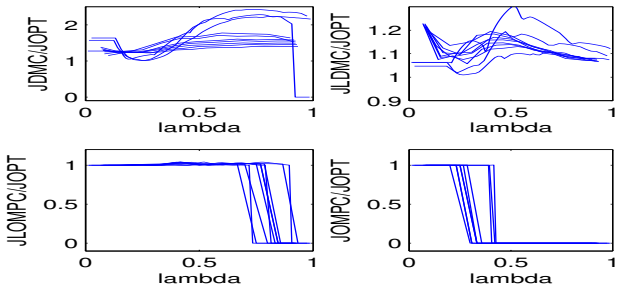

Fig. 1. Ratio of algorithm cost with global optimum for several directions.

B. Example 1

For this example the model is given as:

$$A = \begin{bmatrix} 1.4000 & -0.1050 & -0.1080 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}; B = \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix}; \quad C = \begin{bmatrix} 5 & 7.5 & 0.5 \end{bmatrix}.$$

The constraints are:

$$\overline{u} = 0.04 = -\underline{u}; \quad \overline{\Delta u} = 0.02 = -\underline{\Delta u}; \quad \overline{y} = 1.2 = -\underline{y}.$$

The tuning parameters are $W_u = 10, W_y = 1,\ n_u = n_c = 2,\ a = 0.5$. There are ten directions for the initial states, evenly spread.

Figure 1 shows how the various algorithms performed compared to the global optimum for the various directions as lambda is varied between zero and one. Each direction is denoted by a single curve and infeasibility is denoted by the curve becoming zero.

- A comparison of the top two subplots in figure 1(DMC and LDMC) and in particular the scale on the vertical axis, demonstrates that LDMC has much better performance than DMC but the feasible regions are similar.
- A comparison of the bottom two subplots in figure 1(LOMPC and OMPC) shows performance close to the optimum when feasible, but clearly LOMPC has much better feasibility as the curves do not drop to zero until higher values of lambda. Figure 2 extends this feasibility comparison further by showing the ratio of JLOMPC/JOMPC, and clearly this ratio is close to or less than one when both are feasible and then zero for larger lambda because OMPC has become infeasible.

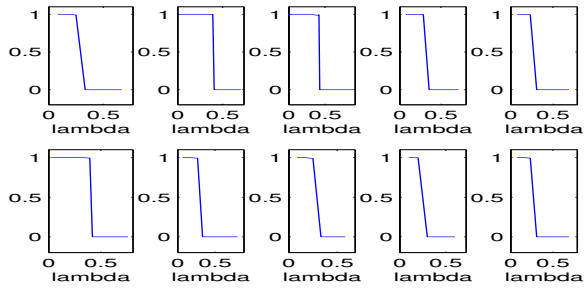

Fig. 2. Ratio of LOMPC cost with OMPC cost for several directions.

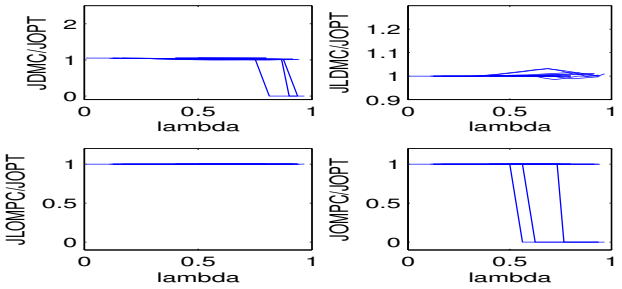

Fig. 3. Ratio of algorithm cost with global optimum for several directions.

C. Example 2

For this example the model is given as:

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

$$B = \begin{bmatrix} 0.0544 & -0.0757 \\ 0.0053 & 0.1477 \\ 0.8647 & 0 \\ 0.5 & 0.2 \end{bmatrix};$$

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

The constraints are:

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

$$\overline{y} = \begin{bmatrix} 7 \\ 1 \end{bmatrix}; \ \underline{y} = \begin{bmatrix} -3 \\ -1 \end{bmatrix}.$$

The tuning parameters are $W_u = 1, W_y = 1$, $n_u = n_c = 3$, $a = 0.5$. There are fifteen directions for the initial states, evenly spread.

Figures 3 and 4 give the same comparisons as explained in figures 1 and 2. Similar conclusions can be derived, although for this example the feasibility and performance benefits are not as significant except in a few directions. A smaller value of the scaling factor a might be better suited.

V. Conclusions

The paper has argued for the potential benefits of Laguerre functions within MPC and more specifically, developed earlier work by integrating these systematically into a dual mode approach, denoted LOMPC. It has been shown by example that this algorithm can outperform a conventional dual mode algorithm by giving substantially better feasibility without compromise to performance. In fact, with example 1, it is notable that for some search directions, one can get feasibility as much as twice as far out with no loss in performance, and this while using exactly the same number of decision variables and optimisation complexity.

We believe the field of how to parameterise the flexibility within the predictions of MPC is understudied, with many authors defaulting to the conventional choice of the explicit values at given sample instants. For our part, the author intends to develop this area further and also look at how the concepts and advantages compare to the recently proposed alternative of clever detuning [3].

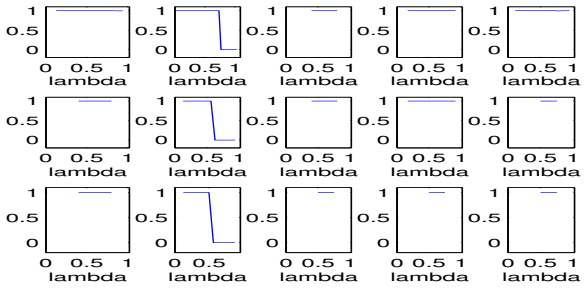

Fig. 4. Ratio of LOMPC cost with OMPC cost for several directions.

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, London, 2003.

[3] Y. Ding and J.A. Rossiter. Compromises between feasibility and performance within linear mpc. In *Proc. IFAC world congress*, 2008.

[4] L. Imsland, J.A. Rossiter, B. Pluymers, and J. Suykens. Robust triple mode mpc. *IJC*, 2008.

[5] B. Kouvaritakis, J.A. Rossiter, and M. Cannon. Linear quadratic feasible predictive control. *Automatica*, 34:1583–1592, 1998.

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

[7] K.R. Muske and J.B. Rawlings. Model predictive control with linear models. *AIChE J.*, 39(2):262–287, 1993.

[8] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428, 1978.

[9] O.J. Rojas and G. C. Goodwin. A simple antiwindup strategy for state constrained linear control. In *IFAC World Congress*, 2002.

[10] J.A. Rossiter. *Model-based predictive control, a practical approach*. Prentice Hall Int., 2003.

[11] J.A. Rossiter, B. Kouvaritakis, and M. Bacic. Interpolation based computationally efficient predictive control. *IJC*, 77:290–301, 2004.

[12] J.A. Rossiter, B. Kouvaritakis, and M. Cannon. An algorithm for reducing complexity in parametric predictive control. *IJC*, 78:1511–1520, 2005.

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

[14] P.O.M. Scokaert and J.B. Rawlings. Constrained linear quadratic regulation. *IEEE Trans AC*, 43(8):1163–1168, 1998.

[15] K.T. Tan and E.G. Gilbert. *Multimode controllers for linear discrete time systems with general state and control constraints*. World Scientific, Singapore, 1992.

[16] L. Wang. Continuous time model predictive control using orthonormal functions. *IJC*, 74(16):1588–1600, 2001.
[17] L. Wang. Discrete model predictive control design using Laguerre functions. *Journal Process Control*, 14:131–142, 2004.

VI. Appendix 1: Laguerre predictions

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

$$L(k+1) = \underbrace{\begin{bmatrix} a & 0 & 0 & 0 & \cdots \\ \beta & a & 0 & 0 & \cdots \\ -a\beta & \beta & a & 0 & \cdots \\ a^2\beta & -a\beta & \beta & a & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{A_L} L(k); \qquad (22)$$

$$L(0) = \sqrt{1-a^2}\begin{bmatrix} 1 \\ -a \\ a^2 \\ -a^3 \\ \vdots \end{bmatrix}; \quad \beta = 1-a^2$$

where $L(k) = [l_1(k)\ l_2(k)\ l_3(k)\ ...\ l_N(k)]^T$ containing the set of Laguerre functions. The dimension of this state space predictor can be taken as large (or small) as needed to capture the desired polynomial sequences. A combination of these sequences up to a horizon n could be computed as

$$\underrightarrow{u}_{k-1} = \underbrace{\begin{bmatrix} L(0)^T \\ L(1)^T \\ \vdots \\ L(n)^T \end{bmatrix}}_{H_L} \eta \qquad (23)$$

where η is the n_L dimension decision variable when one uses the first n_L columns of H_L.

NOTE: because the sequences evolve with a time scaling factor a, it is usually important to ensure that the prediction horizon adopted is large enough to capture all the key dynamics; this for the same reasons practitioners use large output horizons with MPC.

Remark 6.1: For multivariable signals, one can easily modify the above algebra to allow a separate set of sequences for each loop. Hence, it is also easy to allow different a for different loops. This detail is omitted.