Nonlinear model based predictive control using multiple models approach expanded on Laguerre bases

Ghassen MAROUANI, Abdelkader MBAREK, Tarek GARNA and Hassani MESSAOUD
Research Laboratory of Automatic Signal and Image Processing
National School of Engineers of Monastir, University of Monastir
Rue Ibn ELJazzar 5019 Monastir
TUNISIA
Abdelkader.mbarek@enim.rnu.tn

Abstract: This paper proposes a nonlinear model based predictive control (NMPC) algorithm for nonlinear systems by using multiple models approach. To have a less complexity model we expand each linear sub-model on an orthogonal Laguerre basis, the characteristic pole of which should be optimized. In this paper we propose a pole optimization algorithm based on the Gauss-Newton method and we use the provided Laguerre multiple model (LMM) to synthesize a NMPC algorithm. The proposed pole optimization technique as well as the NMPC using LMM approach are validated on a chemical reactor.

Key–Words: Nonlinear systems, Multiple models, Laguerre basis, Predictive control

1 Introduction

Model based predictive control (MPC) is a well-established online control strategy which iteratively computes a control signals by solving an optimization problem over a future time horizon under certain process constraints [1–3]. This optimization uses a prediction model of the future plant behavior. The closed-loop performance depends on the choice of an appropriate model for prediction and several tuning parameters [4] The goal of this work is to synthesize a nonlinear model based predictive control algorithm (NMPC) of nonlinear systems modeled using the multiple models approach proposed in the literature for nonlinear system modeling by Takagi and Sugeno [5], in the context of fuzzy modeling [6]. This approach represents a widespread class as it accurately describes the behavior of a large number of nonlinear systems, although its study can be conducted using theoretical tools developed in the linear framework [7]. The concept of this approach is based on splitting the nonlinear system behavior into a set of operating regions and to describe each operating region by a local linear model or a sub-model. The model describing the whole non linear system behavior is obtained by interpolation variable weights made between various linear models.

In practice, sub-models are obtained by identification, linearization around different operating points or by convex polytopic transformation [8]. In the first situation we can identify, from input/output data, the parameters of the local model corresponding to different operating points. In the second and third situation, it is assumed to have a nonlinear mathematical model [9–12] . In this work, we adopt the black box models that are identified from input/output data around different operating points. The major drawback of the resulting model is its parametric complexity. To overcome this problem, we propose to expand each sub-model on a Laguerre basis that depends on a single, real and stable pole [13,14]. The resulting model entitled Laguerre multiple models (LMM) is a new representation of multiple models approach using Laguerre basis characterized by a parameter vector and a pole vector. In this paper, we proceed to the identification of Laguerre poles and the model coefficients. In a second step we propose and develop a NMPC algorithm using LMM by minimizing a quadratic criterion subject to a set of constraints. The optimization problem is formulated as a quadratic programming (QP) [15].

This paper is organized as follows: in section 2 we remember the principle of multiple model approach. In section 3 we present the Laguerre multiple models obtained from the expansion of FIR multiple models on Laguerre bases. In section 4 we propose the identification procedure of Laguerre multiple models, where we develop a pole optimization algorithm. Section 5 is devoted to develop the NMPC strategy where the unconstrained and constrained cases are treated. Finally, section 6 illustrates the identification procedure given in section 4 by identifying a nonlinear chemical CSTR reactor using the Laguerre multiple models. The proposed NMPC algorithm is also used

to control the output concentration of the CSTR reactor. To raise the efficiency of the proposed algorithm we run it along with an NMPC algorithm build on a GOBF-Volterra model of nonlinear process. The supremacy of the proposed algorithm is confirmed.

2 Principle of multiple models approach

A multiple models is a set of LTI (Linear Time Invariant) and causal sub-models aggregated by an interpolation mechanism to characterize the dynamic behavior of the overall nonlinear system. It is characterized by the number of sub-models, their structure and the choice of weighting functions. A multiple models structure is represented by:

$$y(k) = \sum_{i=1}^{L} \mu_i(\xi(k))\, y_i(k) \tag{1}$$

where, $\hat{y}(k)$ is the multiple models output, L is the sub-model number, $\mu_i(\xi(k))$ is the weighting function associated to the i$^{\text{th}}$ sub-model, $\xi(k)$ is the decision variable and $\hat{y}_i(k)$ is the output of the i^{th} sub-model. The weighting functions $\mu_i(\xi(k))$ allow to determine the relative contribution of each sub-model according to the zone where the system operates and they respect the convexity properties given as follow :

$$\sum_{i=1}^{L} \mu_i(\xi(k)) = 1,\ 0 \leq \mu_i(\xi(k)) \leq 1\ \forall\, i = 1, \cdots, L \tag{2}$$

The weighting functions can be constructed from continuous functions derivatives such as Gaussian functions as follows :

$$\begin{cases} w_i(\xi(k)) = exp\left(-\dfrac{(\xi(k) - c_i)^2}{\sigma_i^2}\right) \\ \mu_i(\xi(k)) = \dfrac{w_i(\xi(k))}{\sum\limits_{j=1}^{L} w_j(\xi(k))} \end{cases} \tag{3}$$

where σ_i and c_i respectively are the dispersion and the centre of the indexed variable $\xi(k)$ and L is the number of sub-models.

3 Laguerre multiple models

In this section we exploit the FIR multiple models in order to develop the Laguerre multiple models (LMM). From relation (1), each sub-model can be described by its output equation $y_i(k)$ given by a finite impulse response (FIR) model as follow:

$$y_i(k) = \sum_{j=0}^{n_i - 1} h^i(j) u(k-j) \tag{4}$$

which can be written in the matrix form as:

$$y_i(k) = \varphi^T(k)\, \theta_i \tag{5}$$

with

$$\begin{cases} \varphi(k) = [u(k)\ u(k-1)\ \ldots\ u(k-n_i+1)]^T, \\ \theta_i = \left[h^i(0)\ h^i(1)\ h^i(2)\ \ldots\ h^i(n_i-1)\right]^T \end{cases} \tag{6}$$

From relation (1) the FIR multiple models is given by:

$$y(k) = \sum_{i=1}^{L} \mu_i(\xi(k))\ \varphi^T(k)\, \theta_i \tag{7}$$

which can be written in matrix form as:

$$y(k) = \Phi^T(k)\, \Theta \tag{8}$$

where Φ is the regression vector given by:

$$\Phi(k) = \left[\mu_1(\xi(k))\varphi^T\ \ldots\ \mu_L(\xi(k))\varphi^T\right]^T \tag{9}$$

and $\Theta \in \mathbb{R}^{n_{pF}}$ is the parameter vector defined as:

$$\Theta = \left[\theta_1^T\ \theta_2^T\ \ldots\ \theta_L^T\right]^T \tag{10}$$

The parameter number of the FIR multiple models is:

$$n_{pF} = \sum_{i=1}^{L} n_i \tag{11}$$

The parameter vector Θ can be estimated using the OLS method.

The major drawback of the FIR multiple models representation is the high number of parameters. To overcome this problem each sub-model is expanded on a Laguerre basis [16]. According to the stability condition of the system in the sense of Bounded Input Bounded Output criterion (BIBO) the coefficients $h^i(j)$ are absolutely summable and they satisfy:

$$\sum_{j=1}^{\infty} \left|h^i(j)\right| < \infty \quad i = 1,\ \ldots,\ L \tag{12}$$

Therefore they belong to the Lebesgue space $\ell^2\,[0,\ +\infty[$. Noting that the orthogonal Laguerre functions form an orthogonal basis in the Lebesgue

space, the coefficients $h^i(j)$ can be then developed on the Laguerre bases $\Im^i = \{\ell_j^i\}_{j=0}^{\infty}$ as follows:

$$h^i(j) = \sum_{n=0}^{\infty} g_n^i \; \ell_n^i(j, \zeta_i), \quad i = 1, \; \ldots, \; L \tag{13}$$

with $\ell_n^i(j, \zeta_i)$ represents the orthogonal functions:

$$\ell_n^i(j, \zeta_i) = \sqrt{1 - \zeta_i^2}\zeta_i^j(-\zeta_i)^n \sum_{f=0}^{min(n,j)} C_f^n C_f^j (\frac{\zeta_i^2 - 1}{\zeta_i^2})^f \tag{14}$$

where $\zeta_i \in]-1, \; 1[$ are the Laguerre poles characterizing the bases $\Im^i$ and g_n^i are the Fourier coefficients $(i = 1, \; \ldots, \; L)$. By replacing $h^i(j)$ given by (13) in the sub-model defined by (4), the resulting sub-model can be written as:

$$y_i(k) = \sum_{j=0}^{n_i - 1} \sum_{n=0}^{\infty} g_n^i \; \ell_n^i(j, \zeta_i) u(k-j), \; i = 1, \; \ldots, \; L \tag{15}$$

Defining $x_n^i(k)$ the output of n^{th} filter:

$$x_n^i(k) = \sum_{j=1}^{\infty} \ell_n^i(j, \zeta_i) \; u(k-j) \tag{16}$$

In practice, the sum in (13) can be truncated to a finite order N_i, we have:

$$h^i(j) = \sum_{n=0}^{N_i - 1} g_n^i \; \ell_n^i(j, \zeta_i), \quad i = 1, \; \ldots, \; L \tag{17}$$

and the sub-model output is given by:

$$y_i(k) = \sum_{n=0}^{N_i - 1} g_n^i \; x_n^i(k), \quad i = 1, \; \ldots, \; L \tag{18}$$

The expansion of all sub-models an Laguerre bases results on a new model titled "Laguerre multiple models (LMM)" given by:

$$y(k) = \sum_{i=1}^{L} \mu_i(\zeta(k)) y_i(k) \tag{19}$$

with $y_i(k)$ given by (18).

Defining the vectors $x^i(k)$ and C_i, $(i = 1, \; \ldots, \; L)$, as:

$$x^i(k) = \left[\; x_0^i(k) \quad x_1^i(k) \quad \cdots \quad x_{N_i - 1}^i(k) \; \right]^T \tag{20}$$

$$C_i = \left[\; g_0^i \quad g_1^i \quad \cdots \quad g_{N_i - 1}^i \; \right]^T \tag{21}$$

we result in the following decoupled state Laguerre multiple models (LMM):

$$\begin{cases} x^i(k+1) = A_i x^i(k) + b_i u(k) \\ y_i = C_i^T x^i(k) \\ y(k) = \sum_{i=1}^{L} \mu_i(\xi(k)) y_i(k) \end{cases} \tag{22}$$

with, for $(r = 1, \; \cdots, \; N_i)$ and $(s = 1, \; \cdots, \; N_i)$

$$A_i(r, s) = \begin{cases} \zeta_i & \text{if } \; r = s \\ (-\zeta_i)^{(r-s-1)}(1 - \zeta_i^2) & \text{if } \; r > s \\ 0 & \text{if } \; r < s \end{cases} \tag{23}$$

and

$$b_i(r) = (-\zeta_i)^{r-1} \sqrt{1 - \zeta_i^2} \cdots, \; N_i) \tag{24}$$

The output $y(k)$ of LMM given by (22) can be written in the vector form as:

$$y(k) = C_m^T X_m(k) \tag{25}$$

where the parameter vector C_m and the state vector $X_m(k)$ are n_{pL}-dimensional and given as follow:

$$C_m = \begin{bmatrix} C_1 \\ \vdots \\ C_i \\ \vdots \\ C_L \end{bmatrix}, \quad X_m(k) = \begin{bmatrix} \mu_1(\xi(k)) x^1(k) \\ \vdots \\ \mu_i(\xi(k)) x^i(k) \\ \vdots \\ \mu_L(\xi(k)) x^L(k) \end{bmatrix} \tag{26}$$

with

$$n_{pL} = \sum_{i=1}^{L} N_i \tag{27}$$

4 Identification of Laguerre multiple models

To identify the Laguerre multiple models, with L sub-models and N truncated order, it is necessary to optimise the poles vector $\zeta_m \in \mathbb{R}^L$.

$$\zeta_m = [\zeta_1, \cdots, \zeta_i, \cdots, \zeta_L] \tag{28}$$

As each Laguerre sub-model is characterized by (N_i) parameters, therefore, we have to identify (n_{pL}) parameters.

4.1 Structural identification

The identification of Laguerre multiple models structure consists on estimating the number L of sub-models and the truncation orders N_i, $(i = 1, \ldots, L)$. We start by identifying the optimal number L of sub-models by setting the truncation orders. Then, we optimize the truncation orders N_i for the optimal number of sub-model. Both optimizations are based on the minimisation of the Normalized Mean Square Error $(NMSE_{dB})$:

$$NMSE_{dB} = 20log_{10}(\frac{\sum_{k=1}^{M}(y_s(k) - y(k))^2}{\sum_{k=1}^{M}(y_s(k)^2)}) \quad (29)$$

where M is the measurement number, $y_s(k)$ is the system output and $y(k)$ is the model output.

4.2 Parametric identification

According to relation (25) we can apply the ordinary least square (OLS) method [17] to identify the parameter vector and consequently the Fourier coefficients of Laguerre sub-models.

$$C_m = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T Y_s \quad (30)$$

where $\mathbb{X}$ is the $(M \times n_{pL})$-dimensional matrix containing the M state vectors and Y_s is the measurement vector:

$$Y_s = [y_s(1), \ldots, y_s(M)]^T \quad (31)$$

and

$$\mathbb{X} = \begin{bmatrix} X_m(1)^T \\ X_m(2)^T \\ \vdots \\ X_m(M)^T \end{bmatrix} =$$

$$\begin{bmatrix} \nu_{11}(x^1)^T(1) & \cdots & \nu_{i1}(x_i)^T(1) & \cdots & \nu_{L1}(x_L)^T(1) \\ \nu_{12}(x_1)^T(2) & \cdots & \nu_{i2}(x_i)^T(2) & \cdots & \nu_{L2}(x_L)^T(2) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \nu_{1M}(x_1)^T(M) & \cdots & \nu_{iM}(x_i)^T(M) & \cdots & \nu_{LM}(x_L)^T(M) \end{bmatrix} \quad (32)$$

where $\nu_{ik} = \mu_i(\xi(k))$, for $(i = 1, ..., L)$ and $(k = 1, ..., M)$, and the $X_m(k)$ vector is given by the relation (26)

4.3 Pole optimization

For the optimization of the poles vector ζ_m given by (28) we execute the Gauss-Newton method as:

$$\zeta_m^{p+1} = \zeta_m^p - (H(\zeta_m) - \lambda I)^{-1}G(\zeta_m) \quad (33)$$

where the gradient $G(\zeta_m)$ and the Hessian matrix $H(\zeta_m)$ are given by:

$$G(\zeta_m) = \frac{\partial J(\zeta_m)}{\partial \zeta_m} \quad (34)$$

$$H(\zeta_m) = \frac{\partial^2 J(\zeta_m)}{\partial \zeta_m \partial \zeta_m^T} = \frac{\partial J(\zeta_m)}{\partial \zeta_m}\frac{\partial J(\zeta_m)}{\partial \zeta_m^T} \quad (35)$$

and $J(\zeta_m)$ is the mean square error defined as:

$$J(\zeta_m) = \frac{1}{2}\sum_{k=1}^{M}\varepsilon(k)^2 = \frac{1}{2}\sum_{k=1}^{M}(y(k) - y_s(k))^2 \quad (36)$$

and its derivative is:

$$\frac{\partial J(\zeta_m)}{\partial \zeta_m} = \sum_{k=1}^{M}(y(k) - y_s(k))\frac{\partial y(k)}{\partial \zeta_m} \quad (37)$$

where:

$$\frac{\partial y(k)}{\partial \zeta_m} = \sum_{i=1}^{L}\mu_i(\xi(k))\frac{\partial y_i(k)}{\partial \zeta_m} \quad (38)$$

with, for $(i = 1, \cdots, L)$:

$$\frac{\partial y_i(k)}{\partial \zeta_m} = \begin{bmatrix} \frac{\partial y_i(k)}{\partial \xi_1} & \frac{\partial y_i(k)}{\partial \xi_2} & \cdots & \frac{\partial y_i(k)}{\partial \xi_r} & \cdots & \frac{\partial y_i(k)}{\partial \xi_L} \end{bmatrix} \quad (39)$$

By using equation (22), we can write for $(i,\ r = 1, \cdots, L)$:

$$\frac{\partial y_i(k)}{\partial \xi_r} = \frac{\partial C_i^T}{\partial \xi_r}x_i(k) + C_i^T\frac{\partial x_i(k)}{\partial \xi_r} \quad (40)$$

The gradient of $x_i(k)$ with respect to ξ_r is:

$$\frac{\partial x_i(k)}{\partial \xi_r} = \frac{\partial A_i}{\partial \xi_r}x_i(k-1) + A_i\frac{\partial x_i(k-1)}{\partial \xi_r} + \frac{\partial b_i}{\partial \xi_r}u(k-1) \quad (41)$$

Since A_i and b_i depend only on ξ_i, therefore:

$$\frac{\partial A_i}{\partial \xi_r} = 0 \text{ et } \frac{\partial b_i}{\partial \xi_r} = 0 \text{ if } i \neq r \quad (42)$$

from relations (41) and (42) and since the initial state of the i^{th} state vector is constant, we conclude that:

$$\frac{\partial x_i(k)}{\partial \xi_r} = 0 \quad \text{if} \quad i \neq r \quad (43)$$

Therefore, the gradient of $y_i(k)$ with respect to ξ_r is written:

$$\frac{\partial y_i(k)}{\partial \xi_r} = \begin{cases} \frac{\partial C_i^T}{\partial \xi_r}x_i(k) + C_i^T\frac{\partial x_i(k)}{\partial \xi_r} & if \ \ i = r \\ \frac{\partial C_i^T}{\partial \xi_r}x_i(k) & if \ \ i \neq r \end{cases} \quad (44)$$

According to (23), the gradient of the matrix A_i is given as follows:

$$\frac{\partial A_i(r,s)}{\partial \zeta_i} = \begin{cases} 1 \text{ if } \quad r = s \\ -(r-s-1)(-\xi_i)^{(r-s-2)}(1-\zeta_i^2) \\ \quad +2(-\zeta_i)^{(r-s)} \text{ if } r > s \\ 0 \quad \text{if } s < r \end{cases} \tag{45}$$

and, according to the relation (24), the gradient of the vector b_i is written:

$$\frac{\partial b_i(r)}{\partial \zeta_i} = -(r-1)(-\zeta_i)^{(r-2)}\sqrt{1-\zeta_i^2} + (-\zeta_i)^{(r-1)}\frac{-\zeta_i}{\sqrt{1-\zeta_i^2}} \tag{46}$$

To compute the gradient of the i^{th} parameter vector C_i with respect to r^{th} pole ξ_r, we compute that of C_m:

$$\frac{\partial C_m}{\partial \xi_r} = \left[\begin{array}{ccccccc} \frac{\partial C_1}{\partial \xi_r} & \frac{\partial C_2}{\partial \xi_r} & \cdots & \frac{\partial C_i}{\partial \xi_r} & \cdots & \frac{\partial C_L}{\partial \xi_r} \end{array} \right] \tag{47}$$

where, $(r = 1, 2, \cdots, L)$
From relation (30) we have:

$$\frac{\partial C_m}{\partial \xi_r} = \frac{\partial \left[\left(\mathbb{X}^T \, \mathbb{X} \right)^{-1} \right]}{\partial \xi_r} \mathbb{X}^T \, y_s + \left(\mathbb{X}^T \, \mathbb{X} \right)^{-1} \frac{\partial \mathbb{X}^T}{\partial \xi_r} y_s \tag{48}$$

or

$$\frac{\partial C_m}{\partial \xi_r} = -\left(\mathbb{X}^T \mathbb{X} \right)^{-2} \left[\frac{\partial \mathbb{X}^T}{\partial \xi_r} \mathbb{X} + \mathbb{X}^T \frac{\partial \mathbb{X}}{\partial \xi_r} \right] \mathbb{X}^T \, y_s + \left(\mathbb{X}^T \, \mathbb{X} \right)^{-1} \frac{\partial \mathbb{X}^T}{\partial \xi_r} y_s \tag{49}$$

where the matrix $\mathbb{X}$ is given by the relation (32) and its gradient with respect to ξ_r is given, using (41) and (43), by:

$$\frac{\partial \mathbb{X}}{\partial \xi_r} = \left[\begin{array}{ccc} 0_{1\times[(r-1)(N+1)]} & \mu_r(\xi(1)) \frac{\partial x_r^T(1)}{\partial \xi_r} & 0_{1\times[(L-r)(N+1)]} \\ 0_{1\times[(r-1)(N+1)]} & \mu_r(\xi(2)) \frac{\partial x_r^T(2)}{\partial \xi_r} & 0_{1\times[(L-r)(N+1)]} \\ \vdots & \vdots & \vdots \\ 0_{1\times[(r-1)(N+1)]} & \mu_r(\xi(M)) \frac{\partial x_r^T(M)}{\partial \xi_r} & 0_{1\times[(L-r)(N+1)]} \end{array} \right] \tag{50}$$

Algorithm 1 : **Pole optimization algorithm**

1. Assume that M couples of input/output are available.
2. Choose the input or output as decision variable $\xi(t)$.
3. Fix the number of sub-model L, the dispersion σ_i and the center c_i $(i = 1, \ldots, L)$.
4. Calculate the weighting functions $\mu_i(\xi(t))$ using equation (3).
5. Fix a threshold Se depending on the desired performance.
6. Choose an initial L-dimensional pole vector ζ_m^p (p=1).
7. Identify the parameter vector $\hat{C}_m$ using the OLS method given by equation (18).
8. Compute the gradient $G(\zeta_m^p)$ given by (34) and the Hessian matrix $H(\zeta_m)$ given by (35) of the quadratic criterion $J(\zeta_m^p)$ with respect to the poles vector ζ_m^p.
9. Calculate the new pole vector ζ_m^{p+1} by relation (33).
10. Calculate the $NMSE_{dB}$ given by (29) for the new pole vector ζ_m^{p+1}.
11. If $(NMSE_{dB} > Se)$
    - p=p+1 ,
    - go to step 7.
12. Else, end of the algorithm

5 NMPC using Laguerre multiple models

5.1 Principle of the predictive control

The Model Based Predictive Control (MPC) is characterized by two essential steps:

- The calculation of the system output prediction $y(k+j/k)$ to determine the process behavior on the prediction horizon $[k+1,\ k+N_p+1]$ depending on inputs applied to the process until the instant $(k-1)T$, the outputs measured until the instant kT and the equations describing the model, T is the sample time.
- The optimisation of a criterion to determine a future control sequence over the control horizon N_u

Only the first element of the optimized sequence is applied to the process.

5.2 The j-step ahead predictor

The j-step ahead predictor on the multiple models approach output $y(k+j/k)$ is calculated from the equation system (22) by multiplying both members of both equations by the polynomial $(1-z^{-1})$, where z^{-1} is the backward shift operator, where:

$$\delta x_i(k) = A_i \delta x_i(k-1) + b_i \delta u(k-1) \tag{51a}$$

$$\begin{aligned} y(k) = \quad & y(k-1) + \sum_{i=1}^{L} \mu_i(\xi(k-1)) C_i^T \delta x_i(k) + \\ & \sum_{i=1}^{L} \left[\mu_i(\xi(k)) - \mu_i(\xi(k-1))\right] C_i^T x_i(k) \end{aligned} \tag{51b}$$

where $\delta u(k)$ and $\delta x_i(k)$ are the control increment and the state vector increment, respectively, defined as :

$$\delta u(k) = u(k) - u(k-1), \quad \delta x_i(k) = x_i(k) - x_i(k-1) \tag{52}$$

The j-step ahead prediction of the output is given from (51b) by:

$$\begin{aligned} y(k+j/k) = y(k+j-1) + \sum_{i=1}^{L} \mu_i(\xi(k+j-1)) C_i^T \delta x_i(k+j) + \\ \sum_{i=1}^{L} \left[\mu_i(\xi(k+j)) - \mu_i(\xi(k+j-1))\right] C_i^T x_i(k+j) \end{aligned} \tag{53}$$

By successive substitutions and taking into account of $y(k/k) = y(k)$, relation (53) leads to:

$$\begin{aligned} y(k+j/k) = y(k) + \sum_{l=1}^{j} \left[\sum_{i=1}^{L} \mu_i(\xi(k+l-1)) C_i^T \delta x_i(k+l) + \right. \\ \left. \sum_{i=1}^{L} \left[\mu_i(\xi(k+l)) - \mu_i(\xi(k+l-1))\right] C_i^T x_i(k+l) \right] \end{aligned} \tag{54}$$

the i^{th} state vector to the $(k+l)^{\text{th}}$ iteration, $x_i(k+l)$, can be written in terms of i^{th} state vector increment $\delta x_i(k)$:

$$x_i(k+l) = x_i(k) + \sum_{m=1}^{l} \delta x_i(k+m) \tag{55}$$

defining:

$$\delta \mu_i(\xi(k+l)) = \mu_i(\xi(k+l)) - \mu_i(\xi(k+l-1)) \tag{56}$$

and the output prediction becomes:

$$\begin{aligned} y(k+j/k) = y(k) + \sum_{l=1}^{j} \sum_{i=1}^{L} \delta \mu_i(\xi(k+l)) C_i^T x_i(k) + \\ \sum_{l=1}^{j} \sum_{i=1}^{L} \mu_i(\xi(k+l-1)) C_i^T \delta x_i(k+l) + \\ \sum_{l=1}^{j} \sum_{i=1}^{L} \delta \mu_i(\xi(k+l)) C_i^T \sum_{m=1}^{l} \delta x_i(k+m) \end{aligned} \tag{57}$$

After some arrangement we obtain:

$$\begin{aligned} y(k+j/k) = y(k) + \sum_{l=1}^{j} \sum_{i=1}^{L} \delta \mu_i(\xi(k+l)) C_i^T x_i(k) + \\ \sum_{i=1}^{L} \sum_{l=1}^{j} \left[\mu_i(\xi(k+l-1)) + \sum_{n=l}^{j} \delta \mu_i(\xi(k+n)) \right] C_i^T \delta x_i(k+l) \end{aligned} \tag{58}$$

where, from (51a) and by successive substitutions the state vector predictor $\delta x_i(k+j)$ is:

$$\delta x_i(k+l) = A_i^l \, \delta x_i(k) + \sum_{m=1}^{l} A_i^{l-m} b_i \, \delta u(k+m-1) \tag{59}$$

By replacing in equation (58), $\delta \hat{x}_i(k+l)$ by its expression given by (59), we obtain:

$$\begin{aligned} y(k+j/k) = y(k) + \sum_{i=1}^{L} \sum_{l=1}^{j} \delta \mu_i(\xi(k+l)) \, C_i^T x_i(k) + \\ \sum_{i=1}^{L} \mu_i(\xi(k+j)) \, C_i^T \left(\sum_{l=1}^{j} A_i^l \right) \delta x_i(k) + \\ \sum_{i=1}^{L} \mu_i(\xi(k+j)) \, C_i^T \sum_{l=1}^{j} \left(\sum_{h=0}^{j-l} A_i^h \right) b_i \, \delta u(k+l-1) \end{aligned} \tag{60}$$

defining:

$$K_{i,n} = \sum_{r=0}^{n} A_i^r \text{ pour } n \geq 0 \text{ et } K_{i,n} = 0 \text{ for } n < 0 \tag{61}$$

equation (60) becomes:

$$\begin{aligned} y(k+j/k) = y(k) + \sum_{i=1}^{L} \sum_{l=1}^{j} \delta \mu_i(\xi(k+l)) \, C_i^T x_i(k) + \\ \sum_{i=1}^{L} \mu_i(\xi(k+j)) \, C_i^T \left[K_{(i,j-l)} - I_{N_i} \right] \, \delta x_i(k) + \\ \sum_{i=1}^{L} \mu_i(\xi(k+j)) \, C_i^T \sum_{l=1}^{j} K_{(i,j-l)} \, b_i \, \delta u(k+l-1) \end{aligned} \tag{62}$$

where I_{N_i} is the N_i-diemsional identity matrix and N_i is the truncation order of the i^{th} sub-model. The j-step

ahead predictor is split into two components, the free and the forced components:

$$y(k+j/k) = y_{fr}(k+j/k) + y_{fo}(k+j/k) \quad (63)$$

where $y_{fr}(k+j/k)$ is the free component determined using the measured outputs till kT and the control up to $(k-1)T$.

$$y_{fr}(k+j/k) = y(k) + \sum_{i=1}^{L}\sum_{l=1}^{j} \delta\mu_i(\xi(k+l))C_i^T x_i(k) + \sum_{i=1}^{L} \mu_i(\xi(k+j))\; C_i^T \left[K_{(i,j-l)} - I_{N_i}\right] \delta x_i(k) \quad (64)$$

and $y_{fo}(k+j/k)$ is the forced component resulting from the action of future control.

$$y_{fo}(k+j/k) = \sum_{i=1}^{L}\left[\mu_i(\xi(k+j))C_i^T \sum_{l=1}^{j} K_{(i,j-l)}\; b_i \delta u(k+l-1)\right] \quad (65)$$

For $j = 1, \ldots, N_p$, relation (63) becomes:

$$Y(k) = Y_{fo}(k) + Y_{fr}(k) \quad (66)$$

where:

$$\begin{cases} Y(k) = \left[\; y(k+1/k) \quad \cdots \quad y(k+N_p/k) \;\right]^T \\ Y_{fr}(k) = \left[\; y_{fr}(k+1/k) \quad \cdots \quad y_{fr}(k+N_p/k) \;\right]^T \\ Y_{fo}(k) = \left[\; y_{fo}(k+1/k) \quad \cdots \quad y_{fo}(k+N_p/k) \;\right]^T \end{cases} \quad (67)$$

The vectors $Y_{fr}(k)$ and $Y_{fo}(k)$ are computed from (64) and (65), respectively, for $(i = 1, \cdots, N_p)$. Then, the prediction vector $Y(k)$ is written in matrix form:

$$Y(k) = G\;\delta U(k) + Y_{fr}(k) \quad (68)$$

where $\delta U(k)$ is the N_p-dimensional future control increment vector.

$$\delta U(k) = [\delta u(k)\; \delta u(k+1)\; \cdots\; \delta u(k+N_p-1)]^T \quad (69)$$

and the N_p-dimensional matrix G is given by:

$$G = \begin{bmatrix} g_{10} & 0 & \cdots & 0 \\ g_{21} & g_{20} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_pN_p-1} & g_{N_pN_p-2} & \cdots & g_{N_p0} \end{bmatrix} \quad (70)$$

from (65), the components $g_{jn}\;\; (n = 0, \ldots, N_p-1)$ and $(j = 1, \ldots, N_p)$ can be written as:

$$g_{jn} = \sum_{i=1}^{L} \mu_i(\xi(k+j))\; C_i^T K_{i,n}\; b_i \quad (71)$$

Taking into account the control horizon N_u the relation (65) becomes:

$$y_{fo}(k+j/k) = \sum_{i=1}^{L}\left[\mu_i(\xi(k))\; C_i^T \sum_{l=1}^{min(j,\,N_u)} K_{(i,j-l)}\; b_i\; \delta u(k+l-1)\right] \quad (72)$$

Consequently the sizes of the matrix G and the vector $\delta U(k)$ are reduced to $(N_p \times N_u)$ and N_u, respectively.

$$G = \begin{bmatrix} g_{10} & 0 & \cdots & 0 \\ g_{21} & g_{20} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{j(N_u-1)} & g_{j(N_u-2)} & \cdots & g_{j0} \\ \vdots & \vdots & \vdots & \vdots \\ g_{N_p(N_p-1)} & g_{N_p(N_p-2)} & \cdots & g_{N_p(N_p-N_u)} \end{bmatrix} \quad (73)$$

and

$$\delta U(k) = [\delta u(k)\; \delta u(k+1)\; \cdots\; \delta u(k+N_u-1)]^T \quad (74)$$

5.3 Control calculation

The control calculation is based on the minimization of the following performance quadratic criterion:

$$J_2(\delta U(k)) = \sum_{j=1}^{N_p} [y(k+j/k) - r(k+j)]^2 + \sum_{j=0}^{N_u-1} \lambda(j)\delta u^2(k+j) \quad (75)$$

where $\lambda(j)$ is a weighting factor generally considered constant and equal to λ and $r(k+j)$ is the reference signal at time instant $(k+j)$.

Two cases will be discussed depending on the consideration or not of physical constraints.

5.3.1 Unconstrained case

The quadratic criterion $J_2(\delta U(k))$ given by (75) can be written in matrix form as:

$$J_2(\delta U(k)) = \|Y(k) - R(k)\|^2 + \delta U(k)^T \Lambda \delta U(k) \quad (76)$$

where Λ is a diagonal weighting matrix, positive definite, whose diagonal elements are λ, $Y(k)$ is the prediction vector given by (67) and $R(k)$ is the N_p-dimensional reference vector given by:

$$R(k) = \left[\begin{array}{ccc} r(k+1) & \cdots & r(k+N_p) \end{array} \right]^T \quad (77)$$

and

$$\Lambda = \lambda I_{N_u} \quad (78)$$

Tacking into account (68), relation (76) becomes:

$$\begin{aligned} J_2(\delta U(k)) &= \|G\delta U(k) + Y_{fr}(k) - R(k)\|^2 + \delta U(k)^T \Lambda\, \delta U(k) \\ &= \delta U^T(k)\, D\, \delta U(k) + 2d^T \delta U(k) + c \end{aligned} \quad (79)$$

with:

$$\begin{array}{c} D = G^T G + \Lambda,\ d = 2G^T(Y_{fr}(k) - R(k)) \text{ and} \\ c = (Y_{fr}(k) - R(k))^T (Y_{fr}(k) - R(k)) \end{array} \quad (80)$$

The minimisation of (79) results in:

$$\delta U(k) = -D^{-1}d = -(G^T G + \Lambda)^{-1} G^T (Y_{fr}(k) - R(k)) \quad (81)$$

Only the first component of $\delta U(k)$ is used to calculate the command $u(k)$ to be applied to the process.

5.3.2 Constrained case

In practice, the processes to be controlled are generally subject to the constraints of the actuators technology, the control system security or the quality desired for the output of the controlled process. MBPC methods allow to take into account some of these constraints explicitly at minimizing the performance criterion J_2, to prevent their violation. These constraints, of inequality type, are generally related to the input / output signal amplitude on the prediction horizon.

$$\delta u_m \leq \delta u(k+j) \leq \delta u_M \ \forall\, j \ \in \{0, \cdots, N_u - 1\} \quad (82)$$

$$u_m \leq u(k+j) \leq u_M \ \forall\, j \ \in \{0, \cdots, N_u - 1\} \quad (83)$$

$$y_m \leq y(k+j) \leq y_M \ \forall\, j \ \in \{1, \cdots, N_p\} \quad (84)$$

Noting that:

$$\begin{aligned} u(k+i) &= \sum_{h=0}^{j} \delta u(k+j-h) + u(k-1) \\ &= \left[\underbrace{1, \cdots, 1}_{j+1} \ \underbrace{0, \cdots, 0}_{N_u - j - 1} \right]^T \delta U(k) + u(k-1) \end{aligned} \quad (85)$$

we have:

$$U(k) = D_1\, \delta U(k) + V_1(k-1) \quad (86)$$

where the N_u-dimensional vectors $\delta U(k)$ and V_1 and square matrix D_1 are defined by:

$$\delta U(k) = \left[\begin{array}{c} \delta u(k) \\ \delta u(k+1) \\ \vdots \\ \delta u(k+N_u-1) \end{array} \right], V_1 = \left[\begin{array}{c} u(k-1) \\ u(k-1) \\ \vdots \\ u(k-1) \end{array} \right]$$
$$\text{and } D_1 = \left[\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 1 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \ldots & 1 \end{array} \right] \quad (87)$$

Replacing the output $y(k+j)$ by their expression predicted $y(k+j/k)$ setting, for $j = (1, \cdots, N_p)$, in the decomposed form (68) the constraint (84) becomes:

$$Y_m \leq G\, \delta U + Y_{fr}(k) \leq Y_M \quad (88)$$

The constraints (82-84) can be written as a function of the future control increment vector $\delta U(k)$:

$$\Gamma \delta U \leq V \quad (89)$$

with :

$$\Gamma = \left[\begin{array}{c} I_{N_u} \\ -I_{N_u} \\ D_1 \\ -D_1 \\ G \\ -G \end{array} \right] \text{ and } V = \left[\begin{array}{c} \delta U_M \\ -\delta U_m \\ U_M - V_1 \\ -U_m + V_1 \\ Y_M - Y_{fr}(k) \\ -Y_m + Y_{fr}(k) \end{array} \right] \quad (90)$$

where the N_u-dimensional vectors U_M, U_m, ΔU_M and ΔU_m are defined as follow:

$$\begin{array}{c} U_M = [u_M \ \cdots \ u_M]^T,\ U_m = [u_m \ \cdots \ u_m]^T \\ \Delta U_M = [\delta u_M \ \cdots \ \delta u_M]^T,\ \Delta U_m = [\delta u_m \ \cdots \ \delta u_m]^T \end{array} \quad (91)$$

and the N_p-dimensional vectors Y_M, Y_m and Y_{fr} are given by:

$$\begin{array}{c} Y_M = [y_M \ \cdots \ y_M]^T \ ,\ Y_m = [y_m \ \cdots \ y_m]^T \\ Y_{fr} = [y_{fr}(k+1/k) \ \cdots \ y_{fr}(k+N_p/k)]^T \end{array} \quad (92)$$

These constraints define a set of control increments as follow:

$$\Psi = \{\delta U(k) \ / \ \Gamma \delta U \leq V\} \quad (93)$$

Taking into account the constraints (82-84) on the input and output signals leads to solving the constraints optimization problem, as follows:

$$\min_{\delta U(k)\in\Psi}(J_2(\delta U(k))) = \min_{\delta U(k)\in\Psi}(\delta U^T(k)D\delta U(k)+ 2d^T\delta U(k)+c) \tag{94}$$

where Ψ is defined by (93).

The Hessian matrix $\nabla^2 J_2(\delta U(k)) = 2D$ of the function $J_2(\delta U(k))$ is positive definite, so that function is strictly convex. Moreover, the constraints (93) defining the convex admissible set Ψ are linear. Therefore, the optimization problem (94) subject to (93) is a convex quadratic programming problem (QP), for which the global solution is unique [18]. Similarly to the unconstrained case, only the first component of $u(k)$ is applied to the system.

Algorithm 2 : NMPC Algorithm using Laguerre multiple models

1. The nonlinear system is modeled by the multi model approach using Laguerre basis where the poles vector ζ_m is optimized by Algorithm 1 and the parameters vector C_m is identified by OLS method.
2. Choose the prediction horizon N_p, the control horizon N_u and determine the N_u-dimensional diagonal weighting matrix Λ.
3. Choose the physical constraints given by (82-84).
4. Compute the N_u-dimensional vectors $\underline{U}_M$, $\underline{U}_m$, $\Delta\underline{U}_M$ and $\Delta\underline{U}_m$ given by (91).
5. Compute the matrices A_i and b_i in terms of $\zeta_m(i)$ for $(i = 1, \ldots, L)$ using equations (23) and (24), respectively.
6. Calculate the matrices G and D_1 given by (73) and (87), respectively.
7. Compute the matrix Γ given by (90).
8. Choose the reference signal $r(k)$ $(k = 1, \ldots, M)$ and initializing the input $u(1)$.
9. for $k = 2$ until $(M - N_p)$
   (a) Apply the inputs $u(k-1)$ to the system and read the output $y(k)$.
   (b) For $(i = 1, \ldots, L)$ calculate $\delta x_i(k)$ given by (51a)
   (c) For $(j = 1, \ldots, N_p)$
   - for $(i = 1, \ldots, L)$ calculate $K_{i,j}$ using relation (61)
   (d) (for $j = 1, \ldots, N_p$) compute $R(k)$ given by (77) and Y_{fr} given by (67).
   (e) Compute the matrix D the vector d and the scalar c given by (80).
   (f) Compute the vector V given by (90)
   (g) Solve the optimization problem given by (94) [15].
   (h) Calculate the control signal $u(k)$.
10. END

6 Application to a CSTR reactor process

6.1 System presentation

The NMPC algorithm proposed in this paper is applied to control the benchmark chemical reactor CSTR [19] whose diagram is given by Figure 1.

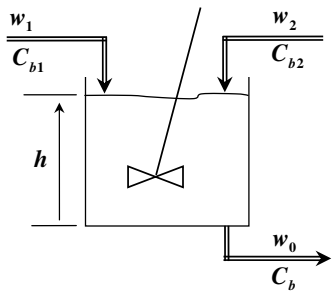

Figure 1: Diagram of the chemical reactor process CSTR

The dynamical model of the system is given by:

$$\frac{\partial h(t)}{\partial t} = w_1(t) + w_2(t) - 0.2\sqrt{h(t)} \tag{95}$$

$$\frac{\partial C_b(t)}{\partial t} = (C_{b1} - C_b(t))\frac{w_1(t)}{h(t)} + (C_{b2} - C_b(t))\frac{w_2(t)}{h(t)} - \frac{k_1 C_b(t)}{(1 + k_2 C_b(t))^2} \tag{96}$$

where $h(t)$ is the liquid level, $C_b(t)$ is the product concentration at the process output, $w_1(t)$ and C_{b1} are respectively the flow rate and the concentration

of product 1, and $w_2(t)$ and C_{b2} are the flow rate and the concentration of product 2. The input concentrations are set to $C_{b1} = 24.9\ Kmol.m{-}3$ and $C_{b2} = 0.15\ Kmol.m{-}3$. The constants associated with the rate of consumption are $k_1 = 1$ and $k_2 = 1$. The CSTR reactor is assumed to be a Single Input Single Output (SISO) system with $w_1(t)$ as input and C_b as output. The flow rate $w_2(t)$ is fixed to $0.7\ L.min{-}1$ and the initial conditions for the CSTR reactor are $h(0) = 0.1m$ and $C_b(0) = 4Kmol.m{-}3$. The simulations are carried out with a sampling interval $T = 1s$. We note that the output C_b is corrupted by an additive white and gaussian noise which is independent of the input signal and which ensures a signal to noise ratio equal to $20dB$.

To apply the proposed NMPC algorithm given by Algorithm 2 to the CSTR process, we start by determining the corresponding Laguerre multiple model which requires structural identification and truncation order by minimizing the NMSE given by (29), parametric identification using the OLS method and poles optimization by the proposed algorithm in this article given by Algorithm 1.

6.2 System identification

In these simulations the weighting functions are chosen as a Gaussian type and the input is considered as a decision variable with a dispersion $\sigma = 0.5$. The CSTR process is identified by the FIR multiple models and the LMM for two and three sub-systems. The identification results are evaluated by the mean square error (MSE) given in Table 1 and Table 2 for FIR multiple models and LMM, respectively.

From both Tables we remark the significant parametric reduction obtained from the expansion of FIR multiple models on Laguerre bases. For example, for the FIR multiple models (Table 1) we obtain a $MSE = -38.20dB$ for three sub-model and sub-model order equal six,(ie $n_{pF} = 18$ parameters). However, we obtain a smaller MSE ($MSE = -49,74dB$) for two sub-model and sub-model order equal two (ie $n_{pL} = 4$ parameters) in the case of LMM (Table 2).

In the flowing, and for applying the NMPC using Laguerre multiple models we can choose, from Table 2, to decompose the CSTR process to two subsystems, then we use two Laguerre sub-models. Then, for have adequate MSE we fixe the truncation order to two. The parameter vector is identified by the OLS method and the poles is optimized by applying algorithm 1. The identification of LMM is summarized in Table 3. In Figure 2 we plot the variation of the poles where we notice the fast convergence of the proposed algorithm, and the optimal pole vector is $\zeta = [0.8108\ 0.4728]$.

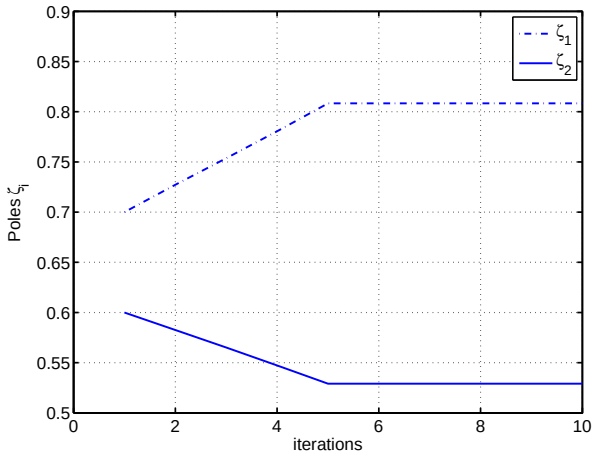

Figure 2: Poles convergence

Thereafter, the CSTR dynamic model given by (95) is modeled by the LMM (22) for a sub-model number $L = 2$ and a truncation order $N = 2$, as follows:

$$\begin{cases} \hat{x}^1(k+1) = A_1\hat{x}^1(k) + b_1u(k) \\ \hat{x}^2(k+1) = A_2\hat{x}^2(k) + b_2u(k) \\ \hat{y}(k) = \mu_1(\xi(k))\hat{C}_1^T\hat{x}^1(k) + \mu_2(\xi(k))\hat{C}_2^T\hat{x}^2(k) \end{cases} \tag{97}$$

where, A_1 and b_1 are calculated from (23) for $\zeta_1 = 0.8108$ and A_2 and b_2 are calculated from (23) for $\zeta_2 = 0.4728$.

To validate the Laguerre multiple models, as well as the identification procedure, we draw in Figure 3 the variation of the CSTR reactor output and the output of the model. We notice the concordance between both outputs.

6.2.1 Control of CSTR using NMPC

The main objective is to maintain the output concentration $C_b(t)$ constant by tuning the flow $w_1(t)$ with $w_2(t) = 0.1$. The tuning parameters are: $N_p = 6$, $N_u = 2$ and $\lambda_i = 0.5$. unconstrained and constrained cases are considered.

Unconstrained case: In Figures 4 and 5 we draw the control and the control increment signals and we note the significant overflows in both signals. These

Table 1: CSTR Process modeled by FIR multiple models

L	2					3				
n_i	2	3	4	5	6	2	3	4	5	6
n_{pF}	4	6	8	10	12	6	9	12		
MSE (dB)	-12.3	-18.2	-28.8	-29.4	-30.1	-20.6	-28.8	-35.4	-37.12	-38.20

Table 2: CSTR Process modeled by Laguerre multiple models

L	2					3				
N_i	2	3	4	5	6	2	3	4	5	6
n_{pL}	4	6	8	10	12	6	9	12	15	18
MSE (dB)	-49.74	-58.25	-65.90	67.28	68.07	-60.57	-70.36	-73.48	-.73.9	-74.1

Table 3: Laguerre multiple models for CSTR process ($L = 2$ and $N_i = 2$)

ξ_{opt}	$\xi_{1,opt} = 0.8108$	$\xi_{2,opt} = 0.4728$
A_i	$A_1 = \begin{bmatrix} 0.8108 & 0 \\ 0.3426 & 0.8108 \end{bmatrix}$	$A_2 = \begin{bmatrix} 0.4728 & 0 \\ 0.7765 & 0.4728 \end{bmatrix}$
b_i	$b_1 = [0.5853 \ -0.4746]^T$	$b_1 = [0.8812 \ -0.4166]^T$
$\hat{C}_m$	$\hat{C}_1 = [0.4380 \ 1.1225]^T$	$\hat{C}_2 = [2.3892 \ 0.6261]^T$

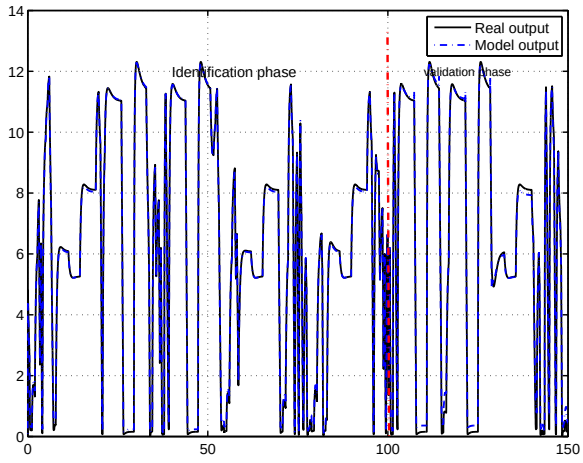

Figure 3: Validation of Laguerre multiple models

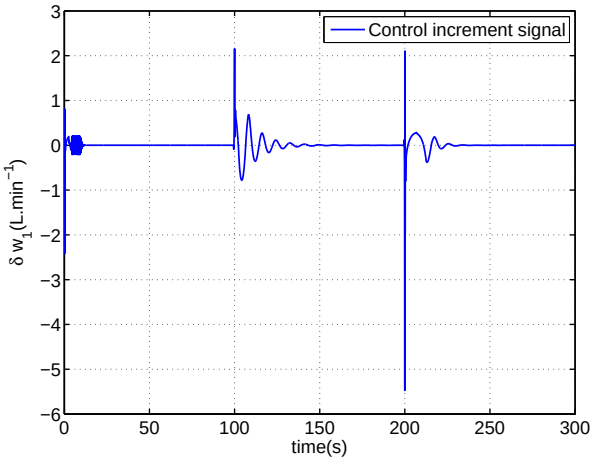

Figure 4: Control increment signal

overflows affect the quality of system output as seen in Figure 6 where overshoot and oscillation characterise the evolution of the output signal.

Constrained case: The physical constraints on input signals are:

$$\begin{cases} 0 \leq u(k+i) \leq 1.33 & i = 0, \cdots, N_p - 1 \\ -0.1 \leq \delta u(k+i) \leq 0.1 & i = 0, \cdots, N_p - 1 \end{cases} \quad (98)$$

Contrarily to the previous case the control and the control increment signal drawn in Figures 7 and 8 generate an output faithful to the reference signal as in Figure 9. From these figures we note the effect of imposing constraints on input signals on the improvement of the system output.

To prove the performances and the effectiveness of the proposed NMPC using LMM algorithm, a comparison was made with the NMPC using GOBF-Volterra model [1]. The CSTR Process is modeld by a second order GOBF-Volterra model for an optimal

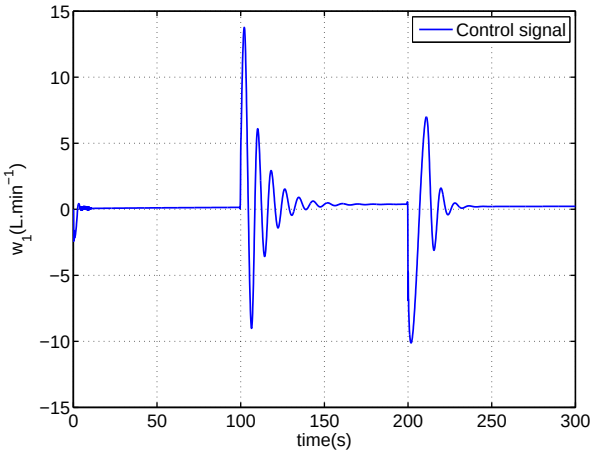

Figure 5: Control signal

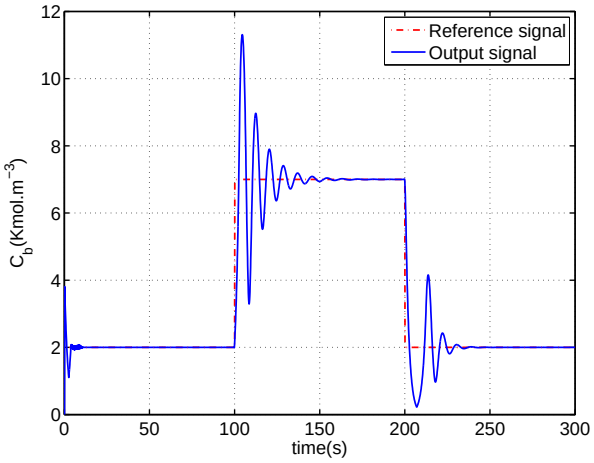

Figure 6: Reference and output signals

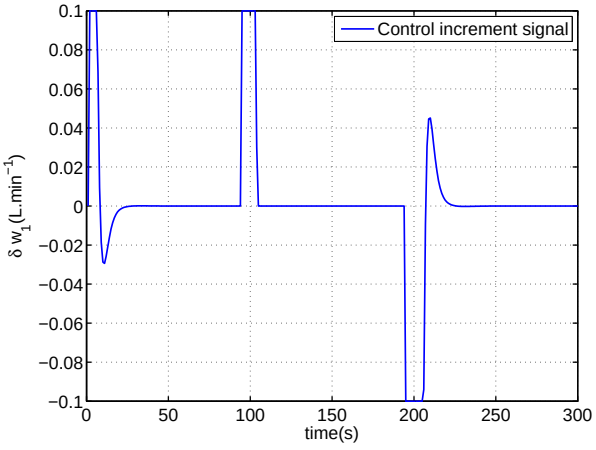

Figure 7: Control increment signal

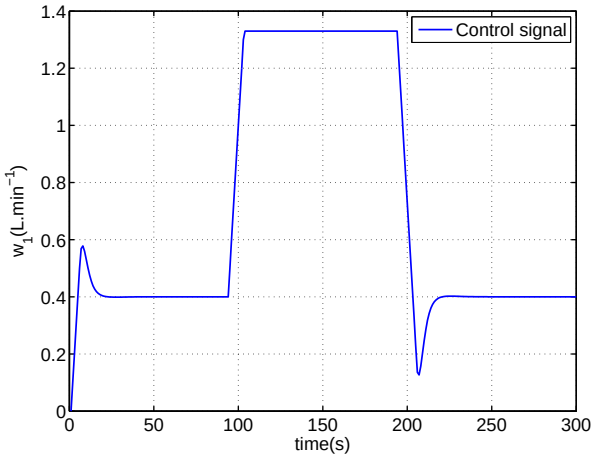

Figure 8: Control signal

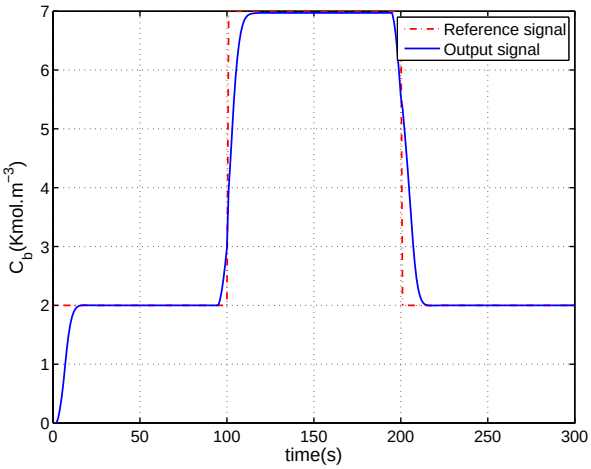

Figure 9: Reference and output signals

truncation order $K_{opt} = 2$. In this case the GOBF-Volterra model is characterized by an optimal pole vector B_{opt} and a parameter vector C given as follow :

$$\begin{cases} B_{opt} = [0.32\ 0.5\ 0.36\ 0.75] \\ C = [2.5\ \ 1.52\ \ 3.2\ 0.75\ \ 0.031] \end{cases} \quad (99)$$

In Figure 10 we draw the reference signal and the CSTR process output by applying the NMPC using LMM and the NMPC using the BOGF-Volterra model. From this Figure we conclude that the NMPC using LMM give better results to control the CSTR process than the NMPC using GOBF-Volterra model. In fact, the latter model output present two deficiencies in terms of precision and oscillation. To improve the results of the NMPC using GOBF-Volterra model it is necessary to increase the parameter number, which may increase the algorithm complexity.

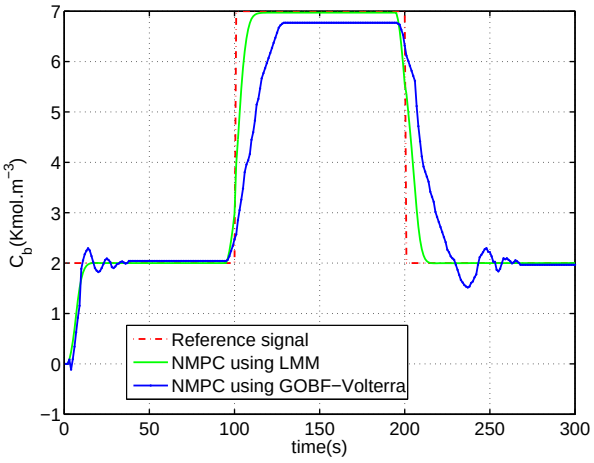

Figure 10: Comparaision between NMPC using LMM and NMPC using GOBF-Volterra model

7 Conclusion

In this paper, we have proposed a new nonlinear predictive control algorithm of nonlinear systems described by decoupled state multiple models approach using Laguerre basis for modelling sub-systems. One more important advantage of the decoupled state multiple model approach is that its structure and the parametric reduction, can facilitate the controller design problem and reduce the computational effort of the NMPC. The optimization problem of NMPC is a minimization of a quadratic criterion with respect to constraints on input signals. The control algorithm is easily solved and yields efficient results. It guarantees the stability of closed-loop system with respect to the choice of the tuning parameters.

References:

[1] K. Bouzrara, A. Mbarek, and T. Garna. Nonlinear predictive controller for uncertain process modelled by gobf-volterra models. *International Journal of Modelling, Identification and Control*, 19(4):307–322, 2013.

[2] D. Li and Y. Xi. The synthesis of robust model predictive control with qp formulation. *Int. J. of Modelling, Identification and Control*, 13(1/2):1–8, 2011.

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

[4] Jean J. Saade and Mohannad Al-Khatib. Performance measures and preferences for fuzzy controllers design algorithms. *Arabian Journal for Science and Engineering*, 29(2B), October 2004.

[5] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to model and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, January 1985.

[6] Wanga C., Kaoa Y., and Yangc G. Exponential stability of impulsive stochastic fuzzy reaction-diffusion cohen-grossberg neural networks with mixed delays. *Neurocomputing*, 89:55–63, July 2012.

[7] Leith D.J. and Leithead W.E. Analytic framework for blended multiple model systems using local linear model. *International Journal of Control*, 72(7-8):605–619, January 1999.

[8] Nagy A. M., Mourot G., Schutz G., and Ragot J. Modelling a biological reactor using a multiple model structure. *Journal Européen des Systèmes Automatisés*, 44(4-5):423–443, January 2010.

[9] Johan A. K. Suykens. *Nonlinear Modeling: Advanced Black-Box Techniques*. Springer-Verlag New York Inc., 2012.

[10] A. Rahrooh and S. Shepard. Identification of nonlinear systems using NARMAX model. *Nonlinear Analysis: Theory, Methods & Applications*, 71(12):1198–1202, 2009.

[11] A. Mbarek, T. Garna, K. Bouzrara, and H. Messaoud. Reduced complexity volterra model of non-linear miso system. *International Journal of Modelling, Identification and Control*, 16(2):134–148, June 2012.

[12] Sala A. and Ario C. Polynomial fuzzy models for nonlinear control: a taylor series approach. *IEEE Transactions on Fuzzy Systems*, 17(6):1284–1295, 2009.

[13] S. Gunnarsson and B. Wahlberg. Some asymptotic results in recursive identification using laguerre models. *International Journal of Adaptive Control and Signal Processing*, 5(5):313–333, 1991.

[14] M. A. Masnadi-Shirazi and N. Ahmed. Optimum laguerre networks for a class of discrette-time systems. *IEEE Transactions on Signal Processing*, 39(9):2104 – 2108, Septembre 1991.

[15] N. Gould and P. L. Toint. Preprocessing for quadratic programming. *Math. Programming*, 100(B):95–132, 2004.

[16] Bouzrara K., Garna T., Ragot J., and Messaoud H. Decomposition of an arx model on laguerre orthonormal bases. *ISA Transactions*, 51(6):848–860, 2012.

[17] L. Ljung. *System identification: Theory for the user*. Prentice Hall PTR, 1999.

[18] R. Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2000.

[19] H. Demuth, M. Beale, and M. Hagan. Neural network toolbox 5. user's guide. Technical report, The MathWorks, 3 Apple Hill Drive Natick, MA 01760-2098, 2005.