Abstract: This paper deals with the identification and the control of nonlinear processes described by input-output models, such as parametric Volterra models. In particular, we extend an adaptive predictive algorithm without taking into account constraints. The calculation of the control law can be posed as a third-order nonlinear program. The building algorithm is based on a new approach using a convolution product. The self-tuning, on-line of the predictive regulator parameters is assured via the weighted recursive least square algorithm WRLS. The nonlinear optimization problem will be reformulated in case of presence of constraints and resolved by a nonlinear programming method as the Lagrange multipliers. The main developed results are applied to an esterification reactor.

Key-Words: Predictive Control- Esterification Reactor - Parametric Volterra Model - Constrained Optimization.

1 Introduction
The successful application of the predictive control on linear processes, notably in the chemical industry and aeronautics, incites researchers to apply this technique to nonlinear processes such as those described by NARMAX, Hammerstein, Wiener, and recently Volterra models. Generally the industrial systems are characterized by a hard nonlinearity, limiting the application of modern control techniques that make recourse of linear model or weakly nonlinear model. It would be, therefore interesting to use nonlinear input-output models as the models of parametric Volterra. Thus, the recourse of NL approach increasingly imposes itself [13, 15, 16].

In literature, many input-output nonlinear models have been proposed for controlling nonlinear systems that follow the MBPC strategies. Yeo and Williams [16] used the MBPC algorithm to control processes described by bilinear models. No approach for identification of their model is presented, and the solution of the optimization problem requires more techniques of the nonlinear programming [1]. Harmsen, and Arkun proposed a nonlinear polynomial model of type (ARMA). The performance of their control scheme was mediocre for various studied cases. Doyle, Haber [2, 14, and 18] used the Second order Volterra model to controls a nonlinear processes. Kimbauer and Jörgi [23] made efficient a parametric Volterra model in a MBPC scheme and they applied it for the control of a distillation column.

In this paper, an adaptive and a modified version of the MBPC strategy proposed by Haber and al [2] is developed and experimentally evaluated. The new adaptive nonlinear controller that allow for the constraints is conceived through direct updating of the incremental prediction equations. When the constraints could be imposed, several MPC methods have been used to solve this problem of controlling nonlinear systems. For the implantation of the MBPC algorithm to a linear system, the problem of the control is reduced to the resolution of a quadratic cost function by linear programming (QP) techniques. However the application of the MBPC algorithm to nonlinear systems introduces a highly nonlinearity, which make impossible to formulate the problem under quadratic format. To reduce the complexity of the NLP technique, many strategies of predictive control have been developed for controlling such a class of nonlinear systems. The basic idea for these
alternative methods is to combine two procedures of different controls: feedback linearization with a standard linear MBPC [21, 22]. The essential difficulty with this approach is caused by the fact that the original optimization problem for the nonlinear system subject to linear constraints on the input has been transformed into an optimization problem for a linear system subject to nonlinear constraints. This second problem is more difficult to solve than the first.

In this work, we consider the original nonlinear MBPC problem and we propose to solve it using constrained nonlinear programming techniques as the Lagrange multipliers [6, 10-12]. The proposed control algorithm is used to control an esterification semi-batch reactor. Usually, original control devices, associated with these reactors are PID regulator, not appropriate where high performances for following precise temperature profiles are desired. Indeed, the dynamics of these reactors are highly nonlinear. Furthermore, the reactant addition at ambient temperature results a unavoidable disruption of the process, whose balancing a real challenge. Another problem associated with these reactors is the stop of the feed stage, always misleads temperature variations.

This paper is organized as follows. In section 2 the declaration of the problem is presented. In section 3 we have developed a non-constrained predictive control algorithm. In section 4 we have developed a recursive estimation algorithm for the identification of the parametric Volterra model parameters. Also an adaptive estimation algorithm for the identification of the control increments [17].

2 Problem Formulations

2.1 Nonlinear process model

The process to control is assumed to be represented by a mono-variable second order parametric Volterra model. This model which is a parametric extension of the Volterra series and that represents judiciously the dynamic nonlinearity for a large class of industrial processes [2, 14, 18, and 20] is given by:

\[ A(q^{-1})y(k) = c_0 + B_1(q^{-1})u(k) + B_2(q_{1}^{-1}, q_{2}^{-1})u^2(k) + \frac{\varepsilon(k)}{\Delta(q^{-1})} \] (1)

where \( y(k) \), \( u(k) \) and \( \varepsilon(k) \) represent respectively the output, the control input, and the system modeling error. \( A(q^{-1}) \) and \( B(q^{-1}) \) are two polynomials of the backward shifting operator \( q^{-1} \) given by:

\[ A(q^{-1}) = 1 + a_1q^{-1} + \ldots + a_{na}q^{-na} ; \]
\[ B_1(q^{-1}) = 1 + b_{11}q^{-1} + \ldots + b_{1mb}q^{-mb} ; \]
\[ B_2(q_{1}^{-1}, q_{2}^{-1}) \] represents the quadratic term of the Volterra model, this quantity is defined by:

\[ B_2(q_{1}^{-1}, q_{2}^{-1})u^2(k) = \sum_{n=0}^{nb} \sum_{m=n}^{nb} b_{2nm}u(k-n)u(k-m) \]

\[ \Delta(q^{-1}) = 1 - q^{-1} ; \] is the difference operator

To alleviate the writing in the rest of this paper, the argument \( (q^j) \) of the polynomials will be omitted. Besides, we consider the following hypotheses:

- The orders \( na, nb \) are already known;
- The coefficients of the polynomials \( A, B1, B2 \) are unknown or/and slightly time-varying.
- The truncation order \( nb \) was chosen to obtain a model with a memory approximately equal to three times the largest time constant of the process [14].

2.2 Optimization criteria

The purpose of the control strategy is to compute future control moves which will minimise some performance function based on the desired output trajectory over a prediction horizon, subject on the input signal. A general mathematical formulation of this problem can be stated as follows [3, 13, 15, 18].

\[
J = E_r \left\{ \sum_{j=1}^{h_c} [y_c(k+j) - \hat{y}(k+j/k)]^2 + \sum_{j=1}^{h_u} \Delta u(k+j-1) \right\}
\]

\[
h_p, h_u \text{ and } \lambda_{u} \text{ are respectively the horizon prediction, the horizon control and the control weighting factor of the control increments [17]. } E_r \{ \} \text{ is the standard deviation operator. } y_c(k+j) \text{ and } \hat{y}(k+j/k) \text{ designate respectively the sequence of future set-points, and the optimal j-step ahead prediction of the system output. Besides we suppose that } h_u = h_p.
\]

\[ \Delta u(k+j-1), \text{ is a sequence of future control increments computed by the optimization problem at time } k: \Delta u(k+j-1) = 0 \forall j \geq h_u. \]

From these criteria, it is clear that the optimization of the cost function requires the calculation of the optimal predictor of the output sequence. We presents in the continuation, two formulations of this predictor. The first one, is according to the control signal, when the second formulation, is based on the control’s increments.

2.3 Non-incremental prediction equation

At instant \((k+j)\) the expression of the system model can be done by the following expression:

\[ Ay(k + j) = c_0 + B_1u(k + j) + B_2u^2(k + j) + \frac{\varepsilon(k + j)}{\Delta(q^{-1})} \]

with \( j=1,2...hp \).

Equation (3) contains the past and future terms of control signal that can be separated by resolution of the following Diophantine equation:
\[ \frac{1}{A} = F_j + q^{-j} G_j \]

\[ F_j = f_0^{(j)} + f_1^{(j)} q^{-1} + \cdots + f_{nj}^{(j)} q^{-nj} \]

\[ G_j = 1 + g_1^{(j)} q^{-1} + \cdots + g_{nj}^{(j)} q^{-nj} \]

(5)

The two polynomials \( F_j \) and \( G_j \) are unique solutions for the previous polynomial identity and have for respective degrees \((j-1)\) and \((na-1)\).

The upper indices \( ^{(j)} \) means that the coefficient depends on the prediction step \( (j) \).

While taking into account the expression (4) and (5), it is easy to approve that the optimal predictor is given by:

\[ \hat{y}(k+j) = c_{op} + G_j y(k) + \beta_1 u(k+j) + \beta_2 u^2(k+j) \]

(6)

with:

\[ \beta_1 = B_1 \cdot F_j \]

\[ \beta_2 = B_2 \cdot F_j \]

\[ c_{op} = c_0 F_j(1) \]

\( \beta_1 \) is a polynomial of degree \((nb+j-1)\),

\( \beta_2 \) is a polynomial matrix of size \((nb+1, nb+j)\).

\( c_{op} = c_0 F_j(1) \) is the constant term.

The symbol \( (\ast) \) represents the convolution product between two polynomials.

### 2.4 Incremental prediction equation

In practice, it is more advantageous to formulate the prediction equation according to the control increments instead of the control signal, thus to formulate it according to the control signals themselves [9].

To simplify the calculation related to the control law synthesis, we consider two control strategies.

#### 1st Strategy:

Only the first control increment is applied to the system and all future increments are supposed equal to zero: \( \Delta u(k) \neq 0, \Delta u(k+1) = \cdots = \Delta u(k+h_u) = 0 \)

#### 2nd Strategy:

The control increments during the control horizon are supposed all equal: \( \Delta u(k) = \Delta u(k+1) = \cdots = \Delta u(k+h_u) \)

For the incremental predictor determination, we divide and multiply the control variables by \( \Delta \), where:

\[ \hat{y}(k+j) = c_{op} + G_j y(k) + \frac{B_1}{\Delta} \cdot F_j \Delta u(k+j) + \frac{B_2}{\Delta^2} \cdot F_j \Delta^2 u^2(k+j) \]

(7)

so, the equation of predictor output is as follows:

\[ \hat{y}(k+j) = c_{op} + G_j y(k) + \frac{B_1}{\Delta} \cdot F_j \Delta u(k+j) + \frac{B_2}{\Delta^2} \cdot F_j \Delta^2 u^2(k+j) \]

(8)

After a Euclidian division of \( F_j \) by \( \Delta \) we will obtain:

\[ F_j = Q_j^{(j)} + q^{-j} R_j^{(j)} \]

(9)

\( Q_j \) is a polynomial of degree \((j-1)\) that:

\[ Q_j = \sum_{i=0}^{j} f_i^{(j)} \]

\[ R_j^{(j)} = 0 \quad \text{if} \quad j = 0, 1, 2, \ldots, j-1 \]

In the same way, we have:

\[ \frac{F_j}{\Delta} = \frac{Q_j^{(j)} + q^{-j} R_j^{(j)}}{\Delta} = \frac{Q_j^{(j)} + q^{-j} (R_j^{(j)} + R_{j-1}^{(j)})}{\Delta} \]

so that:

\[ \frac{F_j}{\Delta^i} = Q_j^{(j)} + q^{-j} R_j^{(j)} \]

with \( R_j^{(j)} = R_{j-1} + R_{j-2} - R_{j+2} q^{-1} \)

and

\[ R_{j-1}^{(j)} = 0 \quad \text{if} \quad j = 0, 1, \ldots, j-1 \]

Finally, the incremental predictive equation can be derived in the following form:

\[ \hat{y}(k+j) = c_{op} + G_j y(k) + Q_j^{(j)} \ast B \ast u(k+j) + Q_j^{(j)} \ast B \ast u^2(k+j) + R_j^{(j)} \ast B \ast u(k+j) \]

(10)

defining thereafter \( \Delta u^*(k) \) as:

\[ \Delta u^*(k+i) = \Delta u(k+i) \quad \text{if} \quad i > 0 \]

\[ \Delta u^*(k+i) = u(k+i) \quad \text{if} \quad i < 0 \]

and we note:

\[ \delta_1^{(j)} = (Q_j^{(j)} + R_j^{(j)} q^{-j}) \ast B_1 \]

\[ \delta_2^{(j)} = (Q_j^{(j)} + R_j^{(j)} q^{-j}) \ast B_2 \]

where \( \delta_1 \) is a polynomial of degree \((nb+j)\) and \( \delta_2 \) is a polynomial matrix of dimension \((nb+j, nb+j)\).

The new expression of the predictor is:

\[ \hat{y}(k+j) = c_{op} + G_j y(k) + \delta_1^{(j)} \Delta u^*(k+j) + \delta_2^{(j)} \Delta u^2(k+j) \]

(11)

This last expression contains the past, present and future increments control terms. The past terms are known at the time of the resolution of the control algorithm. Therefore, we can express prediction equation solely according to the present and future control increments.

\[ \hat{y}(k+j) = v_0^{(j)} + v_1^{(j)} (q^{-1}) \Delta u(k+j) + v_2^{(j)} (q_1^{-1}, q_2^{-1}) \Delta u^2(k+j) \]

(12)

with

\[ v_0^{(j)} = c_{op} + G_j y(k) + \sum_{i=1}^{nb+j-1} \delta_{1i} \Delta u^*(k+j-i) \]

\[ v_1^{(j)} = \sum_{m=j+1}^{nb+j-1} \delta_{2m} \Delta u^2(k+j-i) \]

\[ v_2^{(j)} = \sum_{m=j+1}^{nb+j-1} \delta_{2m} \Delta u^2(k+j-i) \]

where \( m = 1, 2, \ldots, j \) and \( m = 1, 2, \ldots, j \)
3 Control Algorithm
The cost function \( J \) can be written as follows:
\[
J = (v^*_0 + v_1 u + v_2 u^2 + \ldots)^T (v^*_0 + v_1 u + v_2 u^2 + \ldots) + \lambda_u u^T u
\]
(12)
with \( v^*_0 = v_0 - y_c \)
Since only the first control is applied to the process, and assuming strategy 1 or 2, the cost function becomes a function only of the current control increment. The optimization problem becomes one-dimensional equation and takes the following form:
\[
f(u_{op}) = k_0 + k_1 u_{op} + k_2 u_{op}^2 + k_3 u_{op}^3 = 0
\]
(13)
with:
\[
k_0 = (v^*_0 - y_c(k + j)) \left[ \sum_{j=1}^{l} v_{ij} \right]
\]
\[
k_1 = 2 (v^*_0 - y_c(k + j)) \left[ \sum_{j=1}^{l} \sum_{i=1}^{m} v_{ij} \right] + \sum_{j=1}^{l} \sum_{i=1}^{m} v_{ij}^2 + \frac{1}{2} \mu
\]
\[
k_2 = 3 \left[ \sum_{j=1}^{l} v_{ij}^2 \right] \left[ \sum_{j=1}^{l} \sum_{i=1}^{m} v_{ij} \right] + 2 \sum_{j=1}^{l} \sum_{i=1}^{m} v_{ij}^2
\]
\[
k_3 = 2 \left[ \sum_{j=1}^{l} \sum_{i=1}^{m} v_{ij}^2 \right]^2
\]
\( \ell \) is fixed according to the chosen strategy; \( \ell = j \) for strategy 1 and \( \ell = 1 \) for strategy 2.
The equation (13) can be solved with one of the variants of the Newton method as the Beurstow method [17].

4 Adaptive Predictive Control Algorithms
In this section we shall describe an adaptive control strategy, which has a set of techniques, permitting the self-tuning on-line of the regulators parameters, when the process parameters vary in time. The basic idea is to estimate the process model parameters in each estimation step, and construct from the valued parameters a new non-incremental model. Finally we’re going to transform it to another incremental model in each control step. This Prinçipe of control is illustrated by the figure 1.

4.1 Estimation of process parameters
Most nonlinear dynamic system are non-stationary parameters, therefore their behaviour varies in time, through parametric variations. Indeed, for a system describe by the equation (1), can be written as
\[
y(k) = \Theta^T \Phi(k) + \xi(k)
\]
(14)
The parameter estimation vector \( \Theta \) and the regression vector \( \Phi \) are defined as [5]:
\[
\Theta = [v_0, \ldots, a_m, b_{1,0}, \ldots, b_{1,mb}, b_{2,1}, \ldots, b_{2,mb}]^T
\]
\[\Phi(k) = [1, y(k-1), \ldots, y(k-na), u(k-1), \ldots, u(k-nb)]^T\]
The recursive parameter identification of the parametric Volterra model can be obtained via the following weighted recursive last square algorithm with an adjustable forgetting factor [7, 8, 16, 17].
\[
\Theta(k) = \Theta(k-1) + p(k) \Phi(k) + (1 - p(k)) \Theta(k-1)
\]
\[
p(k) = \frac{1}{\alpha(k)} \left[ (p(k-1) - p(k-1) \Phi(k) \Phi^T(k) p(k-1) \right]^T
\]
(15)
\[\alpha(k) = y(k) - \Theta(k-1) \Phi(k)\]
\[\lambda(k) = \lambda_0 (k-1) + \lambda_0^2 (1 - \lambda_0)\]
where \( p(k) \) is the adaptive covariance matrix and \( \lambda (k) \) is a time-varying scalar weighting function (forgetting factor) with \( 0 < \lambda_0 < 1 \) and \( 0 < \lambda_0^2 < 1 \).

4.2 Adaptive control Algorithm
After updating of the model parameters, we calculate the new incremental predictive equation. Indeed, the expression (11) of the predicted output is rewritten:
\[
\hat{y}(k + j) = \hat{v}^0_{ij} + \beta^1_{ij} (q_1^{-1}) \Delta u(k+j) + \beta^2_{ij} (q_2^{-1}) \Delta u(k+j)
\]
(15)
Where \( \hat{v}_0, \hat{v}_1 \) and \( \hat{v}_2 \) are the estimated of the quantities \( v_0, v_1 \) and \( v_2 \). Without constraints the solution of optimization problem leads to a third-degree one-dimensional:
\[
f(u_{op}) = \hat{k}_0 + \hat{k}_1 u_{op} + \hat{k}_2 u_{op}^2 + \hat{k}_3 u_{op}^3 = 0
\]
(16)
The estimates parameter \( \hat{k}_0, \hat{k}_1, \hat{k}_2 \) and \( \hat{k}_3 \) will be calculated in a way like those developed in the section 3. Besides, function (16) is solved analytically.

5 Constrained Adaptive Predictive Control
In this section we’re going to present a generalized structure providing a methodology that takes into account in a systematic way the constraints at the time of the conception and implantation of the control law.

![Adaptive Predictive Control Structure](image)
5.1 Constraints Formulation

In the next of this paper, the first control strategy is applied; the control horizon is maintained equal to the unit, which makes one-dimensional the optimization problem [12, 13, and 18]. Thus the cost function (12) can get under the following form:

\[ g(u) = (v_0^* + v_1 u + v_2 u^2)^2 + \lambda u^2 \]  

(17)

where 

\[ \begin{align*}
  g(u) &= g_0 + g_1 u + g_2 u^2 + g_3 u + g_4 u \\
  \text{subject to} & \quad C_i \bar{u} \leq C_2
\end{align*} \]  

(18)

with

\[ g_0 = (v_1^0 - y_c (k + j))^2 \]

\[ g_1 = 2 (v_1^0 - y_c (k + j)) [v_{ij}^{(j)}] \]

\[ g_2 = 2 (v_1^0 - y_c (k + j)) [v_{ij}^{(j)}]^2 + \lambda u \]

\[ g_3 = 2 [v_{ij}^{(j)}]^2 \]

\[ C_1, C_2 \text{ and } \bar{u} \text{ designed respectively the constraints vectors and the present control increment, they are defined by:} \]

\[ C_1 = [1 \quad -1 \quad 1 \quad -1]^T \]

\[ C_2 = [\bar{u}_{\text{max}} - \bar{u}_{\text{min}} \quad (u_{\text{max}} - u(k-1)) \quad (u(k-1) - u_{\text{min}})]^T \]

where \( u_{\text{max}}, u_{\text{min}}, \bar{u}_{\text{max}}, \bar{u}_{\text{min}} \) are respectively upper limit, lower limit, upper derivative limit, and lower derivative limit of the control input. With constraints the cost function can become be minimised by a one-dimensional search algorithm. The optimization of the control cost function can be simplified, if only the actual control signal would be searched with some assumptions for the future increments. In such cases the minimisation problem becomes one-dimensional as shows the figure 2.

The scheme of the experimental device used in order to valid the theory exposed above is given in figure 3. The core is a 2-1 cylindrical stainless-steel reactor (R) installed in the laboratory of process control at the School of Engineers of Gabes (Tunisia). Its temperature is regulated by means of a fluid circulating through a surrounding jacket. The circulating fluid is assured by the pump (P2). Depending on whether the reactor temperature has to be raised or lowered, the fluid is either heated by a heating changer (El) with a set of three resistors whose electric power can be varied from 0 to 3 kW, or it is cooled in a tubular cooler (E2) who se cooling rate is changed by varying the external cold water flow from 0 to 1200 l/h at most. A vertical stirrer (S), set along the reactor axis, is continuously rotated at a constant speed (usually 600 rpm) in order to keep the medium as homogeneous as possible in both temperature and composition. Should a semi-batch recipe be applied, a specific device (R1), located above the reactor, allows to supply using pump (P1) any additional reactant, the reaction we are interested in is carried at atmospheric pressure. However, in the temperature range under consideration, solvent evaporation always takes place to a significant extent. Accordingly, a water-cooled condenser (C) must be used to condense the solvent in recipient (R2).

A Pt-100 temperature sensor measures the reactor mixture temperature. The inlet (Ti) and outlet (To) temperatures and the flow rate (Fr) of the fluid circulating in the jacket are also measured. The outputs of the transducers used for these acquisitions are connected to a data acquisition-control interface. The reaction carried in this semi batch reactor is an esterification reaction, which is given by the following equation:

\[
\text{R}^1\text{CO}_2\text{H} + \text{R}^2\text{OH} \rightleftharpoons \text{R}^1\text{CO}_2\text{R}^2 + \text{H}_2\text{O}
\]

For identification experiments, and controlling real time we have used the following operated conditions:

- Type of resin: K2411,
- Mass of resin: 50g,
- Initial volume of oil: 750 ml,
- Initial acidity: 35% mass,
- Type of alcohol: Ethanol at 95%,
- Debit of alcohol: 10.5 ml.min-1,
- Period of sampling: 60 s.

For the choice of the exciting input sequence, a uniform signal ranging between 0 and 3.5kW was used to excite the real process.120 data experimental points were collected (Fig.4). This duration of the open-loop experience corresponds to the time needed to complete the reaction furthermore, the average value of mixture temperature for the identification
was fixed at $T = 110^\circ$.

In previous work modelling the esterification reactor, by a set of differential equations, proves that the obtain model is highly nonlinear and very difficult to exploit it for the process control. Therefore, in the rest of the paper, we propose to estimate an appropriate model for the reactor using only input-output information. The identified model will be used for real time controlling this process.

### 6.2 identification of the parametric Volterra model

Open-loop identification tests were first carried out off-line. The main objective of these tests was to estimate the structure identification scheme of the parametric Volterra model. The results of the structure identification scheme with the RELS identification coefficients are given in Table 1.

The necessary conditions of the implementation of the WELS algorithm are taken as:
- The algorithm begins after five sample steps,
- The matrix of covariance $P=10000I$,
- The parameters have for initial values 0.01,
- The forgetting factor was 0.95.

<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1$</td>
<td>$c_0$</td>
<td>-0.0017</td>
</tr>
<tr>
<td>$y(k-1)$</td>
<td>$a_1$</td>
<td>1.0294</td>
</tr>
<tr>
<td>$y(k-2)$</td>
<td>$a_2$</td>
<td>-0.0212</td>
</tr>
<tr>
<td>$u(k-1)$</td>
<td>$b_1$</td>
<td>-0.0348</td>
</tr>
<tr>
<td>$u(k-2)$</td>
<td>$b_2$</td>
<td>0.0065</td>
</tr>
<tr>
<td>$u(k-1)^2$</td>
<td>$b_{11}$</td>
<td>0.0209</td>
</tr>
<tr>
<td>$u(k-1)u(k-2)$</td>
<td>$b_{12}$</td>
<td>0.0051</td>
</tr>
<tr>
<td>$u(k-2)^2$</td>
<td>$b_{22}$</td>
<td>0.0035</td>
</tr>
</tbody>
</table>

Table 1 Parameters of the parametric Volterra model

### 6.3 Closed-loop control

The implantation of the unconstrained NLPC algorithm, to an esterification reactor illustrates their validity for industrial process regulation (Fig. 5).

Practically the control sequence to apply for an industrial process is not gratis it’s limited by constraints on the input signal, and/or on the control increments, or two together. Figure 6 show the estimated temperature adjusted around $110^\circ$C, the measured temperature, the control signal and the control increment under...
constraints. The chosen constraints are imposed for this application as follows: \( 1650W \leq P \leq 2000W \) for power and \( |\Delta P| \leq 50W \) for power increments (for times \( t \in [50, 150] \) mn).

The tuning parameters for NLPC algorithm are the following: the prediction horizon \( h_p = 6 \), the control horizon \( h_u = 1 \) and the weighting factor of the control increments \( \lambda_u = 0.1 \).

Fig.6.a. Closed-loop responses for constrained NLPC

Fig.6.b. Control low generate by NLPC algorithm

7 Conclusions
In this paper, a formulation of a nonlinear predictive control scheme based on the second-order parametric Volterra model, using the control increments was presented. The updating of the incremental prediction equation is assured via the WRLS algorithm. This formulation determines the control action by directly solving a third order nonlinear programming problem. If there are input constraints, then a constrained optimization problem can be presented, the optimal control increments can be computed numerically. The developed controller yielded improved performance for the control of an esterification reactor that we look for adjust its temperature about 110°C. The presented experimental results show the good performance of the developed controls strategy. Besides these experiences show the efficiency, flexibility and the good performance of the NLPC algorithm applied for the control of the reactor temperature.

References:


