Implementations of a Hammerstein Fuzzy-Neural Model for Predictive Control of a Lyophilization Plant

Yancho Todorov, *IEEE member*
Institute of Cryobiology and
Food Technologies,
Sofia, Bulgaria,
yancho.todorov@ieee.org

Sevil Ahmed, Michail Petrov
Technical University-Sofia,
Branch Plovdiv,
Plovdiv, Bulgaria,
sevil.ahmed@gmail.com,
mpetrov@tu-plovdiv.bg

Vasilliy Chitanov
Central Laboratory of
Applied Physics,
Bulgarian Academy of Sciences,
Plovdiv, Bulgaria
vchitanov@gmail.com

Abstract—**This paper describes two methodologies for implementation of Hammerstein model by using different input-output representations into model predictive control schemes. The model nonlinearity is easily approximated using a simple Takagi-Sugeno inference, while the linear parts are flexibly introduced. As optimization procedures for predictive control are used a standard gradient optimization method and an implementation of Hildreth Quadratic Programming. A comparison between the proposed control strategies is made by simulation experiments for control of nonlinear lyophilization plant.**

Keywords- predictive control, fuzzy-neural models, optimization, gradient descent, Hildreth quadratic programming, lyophilization

I. INTRODUCTION

Many biopharmaceutical entries are not stable for long periods of time as an aqueous solution. In the presence of water as solvent, such materials either degrade through a chemical reaction of hydrolysis or undergo other chemical reactions allowed by the molecular mobility provided in a liquid state. In addition these degradation reactions may be accelerated in higher temperatures. Low temperature vacuum-drying conditions afforded by lyophilization provide safe and effective mechanism for removing the solvent and converting the material to a solid state thereby promoting long-term stability [1].

Nowadays, pharmaceutical industries are generating many products each year, thus creating pressure for reliable determination and control of the drying cycles during lyophilization. The lyophilization is the *most* complex and *expensive* form of drying, because of the amount of the consumed energy during the process batch; a factor which one would like to reduce by advanced control methodologies, such as Model Predictive Control (MPC). MPC has recently found a wide acceptance in industrial applications, where dynamics are relatively slow and hence can accommodate on-line optimization easily [2]. During the last years many researchers report different suitable applications of MPC in lyophilization [3-6].

One of the most frequently studied class of nonlinear models used in MPC are the so-called block oriented nonlinear models [7], which consist the interconnection of linear dynamic systems and static nonlinearities. Within this class, two of the more common model structures are the Hammerstein and the Wiener models. The Hammerstein model consist a cascade connection of static nonlinearity followed by a linear dynamic system [8] and it is successfully applied for nonlinear system representation in different applications [9-11].

This paper represents a comparative study between explicit and non-explicit MPC implementations by using a simple Fuzzy-Neural (FN) Hammerstein model, based on the Takagi-Sugeno fuzzy-neural technique. The possible benefits of the proposed approaches are demonstrated by simulation experiments in MATLAB environment to control the temperature of the heating shelves in notion to the temperature of the ice front during a freeze-drying cycle.

II. FUZZY-NEURAL HAMMERSTEIN MODELS

A. Fuzzy-neural NARX Hammerstein model

Using a simple Takagi-Sugeno inference, the static nonlinearity into the Hammerstein model can be easily represented as [12]:

$$u_{m} = f_u(v(k)), \quad \text{where } v(k) = [u(k),....,u(k-n_u)] \tag{1}$$

The unknown nonlinear functions f_u can be approximated by Takagi-Sugeno type fuzzy rules:

$$\begin{gathered} R^{(i)}: if\ v_1\ is\ \widetilde{A}_1^{(i)}\ and\ v_p\ is\ \widetilde{A}_p^{(i)}\ then\ f_u^{(i)}(k) \\ f_u^{(i)}(k) = d_1^{(i)}u(k) + d_2^{(i)}u(k-1) + \ldots + d_{n_u}^{(i)}u(k-n_u) + d_0^{(i)} \end{gathered} \tag{2}$$

The upper index is: *(i)=1, 2,…N*, where *N* is the number of the fuzzy rules, A_i is an activated fuzzy set defined in the universe of discourse of the input $v=[v_{1,}\ v_{2,\ldots}v_p]$ and the crisp coefficients d_{ji}, are the coefficients into the Sugeno function

f_u. From a given input vector, the output of the fuzzy model is inferred by computing the following equation:

$$u_m(k) = f_u{}^{(i)}(k) g_u{}^{(i)} \quad \text{where} \quad g_u{}^{(i)} = \prod_{i=1}^{N} \mu_{ui}{}^{(i)} \tag{3}$$

where μ_{ui} are the degrees of fulfillment in notion to ui-th activated fuzzy membership function. Afterwards, the linear part is introduced into the fuzzy model as follows:

$$\begin{aligned} f_s{}^{(i)}(k) &= a_1{}^{(i)} y(k-1) + \ldots + a_{n_y}{}^{(i)} y(k-n_y) + \\ &+ b_1{}^{(i)} u_m(k) + \ldots + b_{n_u}{}^{(i)} u_m(k-n_u) + b_o \end{aligned} \tag{4}$$

In general, the designed FN Hammerstein model can be represented as:

$$\begin{aligned} y(k+1) &= \sum_{j=1}^{N}\left(\sum_{i=1}^{n_a} a_i y(k-i+1) + \sum_{i=1}^{n_b} b_i u_{mj}(k-i-n_d+1)\right) \\ &= \sum_{j=1}^{N}\left(\sum_{i=1}^{n_a} a_i y(k-i+1) + \sum_{i=1}^{n_b} b_i d_j g_{uj}(u(k-i-n_d+1))\right) \end{aligned} \tag{5}$$

B. Fuzzy-neural State-Space Hammerstein model

Using the above notations, the general states-space FN Hammerstein model can be expressed as:

$$\left| \begin{aligned} & x_1(k+1) = f_x(x_1(k), u(k)) \\ & z(k) = f_z(x_1(k), u(k)) \end{aligned} \right. \tag{6}$$

where *$x_1(k)$*, *$u(k)$* and *$z(k)$* are vectors for the state, the input and the output of the nonlinear part. The unknown nonlinear functions f_x and f_z can be approximated by Takagi-Sugeno type fuzzy rules:

$$\begin{aligned} & R^{(i)}\!: if\; r_1(k)\; is\; M_1{}^{(i)} \ldots and \ldots r_p(k)\; is\; M_p{}^{(i)} \\ & then \quad \left| \begin{aligned} & x_1(k+1) = A_1 x_1(k) + B_1 u(k) \\ & z(k) = C_1 x_1(k) + D_1 u(k) + \vartheta \end{aligned} \right|^{(i)} \end{aligned} \tag{7}$$

where R is the i^{th} rule of the rule base, r_p is the corresponding state regressor, M_i is a membership function of a fuzzy set, $A^{(i)}$, $B^{(i)}$, $C^{(i)}$ and $D^{(i)}$ are state-space matrices according each rule and ϑ is a vector of free elements (offsets). The role of the model offsets is to compensate the possible disturbances in the process [13]. For each input vector, the output of the fuzzy model is computed by using the following equation:

$$\left| \begin{aligned} & x_1{}^{(i)}(k+1) = f_x{}^{(i)} g_u{}^{(i)} \\ & z^{(i)}(k) = f_z{}^{(i)} g_u{}^{(i)} \end{aligned} \right. \quad \text{where } g_u{}^{(i)} = \prod_{i=1}^{N} \mu_{ui}{}^{(i)} \tag{8}$$

where μ_{ui} are the degrees of fulfillment in notion the activated fuzzy membership function. Thereafter, the linear part is introduced as:

$$\begin{aligned} x_2(k+1) &= A_2 x_2(k) + B_2 z(k) \\ y(k) &= C_2 x_2(k) + D_2 z(k) + \vartheta_1 \end{aligned} \tag{9}$$

In notion to each activated fuzzy rule, the general local fuzzy model can be expressed as:

$$\begin{aligned} & \begin{vmatrix} \dot{x}_1 \\ \dot{x}_2 \end{vmatrix} = \begin{vmatrix} A_1{}^{(i)} & 0 \\ B_2 C_1{}^{(i)} & A_2 \end{vmatrix} \times \begin{vmatrix} x_1 \\ x_2 \end{vmatrix} + \begin{vmatrix} B_1{}^{(i)} \\ B_2 D_1{}^{(i)} \end{vmatrix} u + \begin{vmatrix} 0 \\ B_2 \vartheta^{(i)} \end{vmatrix} \\ & y = \begin{vmatrix} D_2 C_1{}^{(i)} & C_2 \end{vmatrix} \times \begin{vmatrix} x_1 \\ x_2 \end{vmatrix} + D_1{}^{(i)} D_2 u + \left| D_2 \vartheta^{(i)} + \vartheta_1 \right| \end{aligned} \tag{10}$$

Finally, the general model representation is expressed as:

$$\left| \begin{aligned} & x_1(k+1) = \sum\nolimits_{i=1}^{N} g_{u\,i}(A_1 x_1(k) + B_1 u(k)) \\ & x_2(k+1) = \sum\nolimits_{i=1}^{N} (A_2 x_2(k) + B_2 g_{u\,i}(C_1 x_1(k) + D_1 u(k) + \vartheta) \\ & y(k) = \sum\nolimits_{i=1}^{N} (C_2 x_2(k) + D_2 g_{u\,i}(C_1 x_1(k) + D_1 u(k) + \vartheta) + \vartheta_1 \end{aligned} \right. \tag{11}$$

C. Learning algorithm for the designed fuzzy-neural models

The identification procedure involves the determination of two groups of parameters: Gaussian membership functions and linear coefficients in the rules premise and consequent parts, respectively. The designed algorithm minimizes an instant error measurement function *$E(k)=\varepsilon^2/2$* and *$\varepsilon(k)=y(k)-\hat{y}(k)$*, between the real plant output *$y(k)$* and this one estimated by the fuzzy-neural model *$\hat{y}(k)$*. A two step gradient learning procedure is performed to calculate the current values of the parameters.

Assuming that β_{si} is an adjustable s^{-th} coefficient for the functions f_x, f_z and f_u into the i^{th} activated rule as a connection in the output neuron, the general parameter learning rule for the consequent parameters is: *$\beta(k+1)=\beta(k)+\eta(\partial E/\partial \beta_i)$*. After calculating the partial derivatives, the final recurrent predictions for each adjustable coefficient β_{si} *(a(i), b(i))* or *d(i))* for the NARX model and for the State Space model (*a(i), b(i), c(i), d(i) or ϑ(i)*) are obtained by using the same equations:

$$\begin{aligned} \beta_{si}(k+1) &= \beta_{si}(k) + \eta\varepsilon(k)\overline{\mu}^{(i)} x_i(k) \\ \beta_{0i}(k+1) &= \beta_{0j}(k) + \eta\varepsilon(k)\overline{\mu}^{(i)} \end{aligned} \tag{12}$$

The output error E can be used back directly to the input layer, where the premise adjustable parameters are situated (center – c_{li} and the deviation – σ_{li} of a Gaussian fuzzy set). Through the interconnections of the corresponding membership degrees where the link weights are unit, the error E is propagated. Thus, the learning procedure for the rule premise parameters can be expressed by the same learning rule:

$$c_{li}(k+1) = c_{li}(k) + \eta\varepsilon(k)\bar{g}_{ui}(k)[f^{(i)}(k) - \hat{y}(k)]\frac{[r_l(k) - c_{li}]}{\sigma_{li}{}^2(k)}$$
$$\sigma_{li}(k+1) = \sigma_{lj}(k) + \eta\varepsilon(k)\bar{g}_{ui}(k)[f^{(i)}(k) - \hat{y}(k)]\frac{[r_l(k) - c_{li}]^2}{\sigma_{li}{}^3(k)} \qquad (13)$$

III. MODEL PREDICTIVE CONTROL STRATEGIES

A. Solution of the non-explicit optimization problem

Using the designed NARX type FN model, the *Optimization Algorithm* computes the future control actions at each sampling period, by minimizing the following cost function:

$$J(k,u(k)) = \sum_{i=N_1}^{N_2} \left\| r(k+i) - \hat{y}(k+i) \right\|^2 + \rho \sum_{i=N_1}^{N_u} \left\| \Delta u(k+i-1) \right\|^2 \qquad (14)$$

where $\hat{y}$ is the predicted model output, r is the reference and Δu is the rate change of the control action. The tuning parameters of the predictive controller are: N_1 -the minimum prediction horizon, N_2 -the maximum prediction horizon, N_u -the control horizon and ρ -the weighting factor penalizing changes in the control actions. Taking into account, that the above criterion is quadratic one and there are no constraints, in respect to future control actions a simple analytic minimization can be performed, by applying the condition $\Delta J=0$. [12,14]. The partial derivatives $\partial Y(k)/\partial U(k)$ can be calculated by the following equations using the designed model:

$$\frac{\partial \hat{y}(k)}{\partial u(k)} = \sum_{i=1}^{N} b_1^{(i)} \bar{g}_u^{(i)}(k)$$
$$\frac{\partial \hat{y}(k+N_2)}{\partial u(k)} = \sum_{i=1}^{N} \begin{bmatrix} a_1^{(i)} \dfrac{\partial \hat{y}(k+N_2-1)}{\partial u(k)} + \ldots \\ + a_{ny}^{(i)} \dfrac{\partial \hat{y}(k+N_2-2)}{\partial u(k)} \end{bmatrix} \bar{g}_u^{(i)}(k+N_2) \qquad (15)$$

The second group of partial derivatives is given by: $\partial U(k)/\partial U(k)$. Since, $\Delta u(k)=u(k)+u(k-1)$, then this group of derivatives represent a simple matrix with zeros and ones.

Finally, the control actions are calculated iteratively according the following expressions:

$$\frac{\partial J[k,U(k)]}{\partial u(k)} = -2\begin{bmatrix} \hat{e}(k+N_1)\dfrac{\partial \hat{y}(k+N_1)}{\partial u(k)} + \ldots \\ + \hat{e}(k+N_2)\dfrac{\partial \hat{y}(k+N_2)}{\partial u(k)} \end{bmatrix} + $$
$$+2\rho\Delta u(k) - 2\rho\Delta u(k+1) \qquad (16)$$
$$\frac{\partial J[k,U(k)]}{\partial u(k+N_u-1)} = -2\begin{bmatrix} \hat{e}(k+N_1)\dfrac{\partial \hat{y}(k+N_1)}{\partial u(k+N_u-1)} + \ldots \\ + \hat{e}(k+N_2)\dfrac{\partial \hat{y}(k+N_2)}{\partial u(k+N_u-1)} \end{bmatrix} + $$
$$+2\rho\Delta u(k+N_u-1)$$

B. Quadratic Programming procedure

Using the designed FN state-space Hammerstein model, the *Optimization Algorithm* computes the future control sequence, by minimizing the following cost:

$$J(k,u(k)) = \sum_{i=N_1}^{N_2} \left\| \hat{y}(k+i) - r(k+i) \right\|^2 Q + \sum_{i=N_1}^{N_u} \left\| \Delta u(k+i) \right\|^2 R \qquad (17)$$

subject to $\Omega\Delta\mathrm{U} \le \gamma$

which can be simply expressed in vector form as: $J(k)=[Y(k)-T(k)]^2Q+[\Delta U(k)]^2R$; where, Y is the matrix of the predicted plant output, T is the reference matrix, ΔU the matrix of the predicted rate of the controls and Q and R are the matrices, penalizing the changes in error and control term of the cost function. Taking into account the general prediction form of a linear state-space model, the predicted output can be derived as $Y(k)=\Psi X(k)+Yu(k-1)+\Theta\Delta U(k)+\vartheta$ [15].

Then, the predicted error can be defined as: $E(k)=T(k)-\Psi X(k)-Yu(k-1)-\vartheta$ and if $\Delta U=0$ is assumed, using the last notation, the cost function can be rewritten as: $J(k)=\Delta U^T H\Delta U+\Delta U^T\Phi+E^TQE$, where $\Phi=-2\Theta^TQE(k)$ and $H=\Theta^TQ\Theta+R$. Differentiating the gradient of J with respect to ΔU, gives the matrix of second order derivatives, or Hessian of $J(k)$: $\partial^2 J(k)/\partial\Delta U^2(k)=2H=2(\Theta^TQ\Theta+R)$. If $Q(i)\geq 0$ for each i, this ensures that $\Theta^TQ\Theta\geq 0$. So, if $R\geq 0$ then the Hessian is certainly positive-definite, this is enough to guarantee that the minimum has been reached.

Linear constraints usually take place in the quadratic programming. Since, $U(k)$ and $Y(k)$ are not explicitly included in the optimization problem, the constraints can be expressed in terms of ΔU, where the first row represents the constraints on the amplitude of the control signal, the second one the constraints on the output changes and the last the constraints on the rate change of the control signal [15].

$$\begin{bmatrix} F_1 \\ G\Theta \\ W \end{bmatrix} \Delta U \le \begin{bmatrix} -F_2 u(k-1) + f \\ -G(\Psi X(k) + \Upsilon u(k-1)) + g \\ w \end{bmatrix} \qquad (18)$$

C. Hildreth Quadratic Programming procedure

Necessary condition for optimization in presence of inequality constraints is the satisfaction of the Kuhn-Tucker conditions, where the vector λ contains the Lagrange multipliers. These conditions can be expressed simply using the active set notation.

Using the active set procedures which belong to the family of primal methods, the solutions depend on the primal decision variables and the computational burden is quite large, if many constraints are imposed. On the other hand, programming of an active set method is a difficult task. For this purpose, a dual method is applied in order to identify the constraints which are not active, so they can be eliminated from the solution [16]. Thus, a simple programming task for finding the optimal solutions of the constrained minimization problem can be adopted. The dual is also a quadratic programming problem with λ as the decision variable: $\Delta U=-H^{-1}\Phi-H^{-1}(\Omega_{act}\lambda_{act})$, min $(\lambda^T M\lambda+\lambda^T K+\gamma^T M^{-1}\gamma)$ and $\lambda>0$.

A simple Hildreth quadratic programming procedure is used for solving this dual problem, where the direction vectors were selected to be equal to the basis vectors. Then, the λ vector is varied one component at a time. At a given stage, once obtained the vector $\lambda \geq 0$, the attention is focused on a single component λ_i, which is further adjusted to minimize the objective function. If that requires $\lambda_i<0$, λ_i can be set equal to zero, otherwise the objective function is decreased. Then, the next component λ_i+1, is considered. After one complete cycle through the components as iteration taking the vector λ^m to λ^{m+1}, the method can be expressed explicitly as: $\lambda_i^{m+1}=max\ (0,\ \alpha_i^{m+1})$ where:

$$\alpha_i^{m+1}=-\frac{1}{h_{ii}}\left[k_i+\sum_{j=1}^{i-1}h_{ij}\lambda_j^{m+1}+\sum_{j=i+1}^{n}h_{ij}\lambda_j^{m}\right] \quad (19)$$

where the scalar h_{ij} is the ij^{th} element in the matrix M, and k_i is the i^{th} element in the vector K. Also, there are two sets of λ values in the computation: one involves λ_m and one involves the updated λ_{m+1}. Because the converged λ vector contains either zero or positive values of the Lagrange multipliers, finally the expression has the form: $\Delta U=-H^{-1}(\Phi+\Omega^T\lambda^*)$ where: $\lambda^*_{act}=-(\Omega_{act}H^{-1}\Omega_{act}^{T})^{-1}(\gamma_{act}+\Omega_{act}H^{-1}\Omega_{act}^{T})$

IV. SIMULATION EXPERIMENTS

A. Experimental plant description

On Fig. 1 a schematic diagram of the components of a lyophilization apparatus is shown. It consists of a drying chamber (1); temperature controlled shelves (2), a condenser (3) and a vacuum pump (4). The major purposes of the shelves are to cool/freeze or to supply heat to the product by the corresponding heating or refrigeration system (5). The product is placed on supportive product shelves (6) and the chamber is isolated from the condenser by the valve (7). The vacuum system is placed after condenser. The sublimation driving force pipes out the sublimed water from the product, which transforms back to ice on the condenser wall.

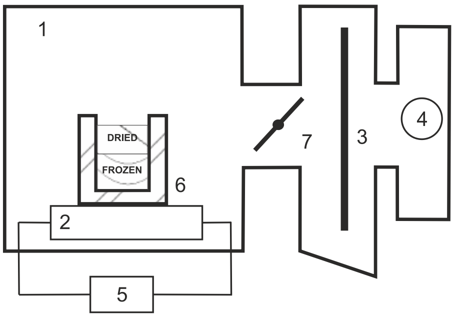

Figure 1. Shematic diagram of a lyophilization plant.

The product is loaded on the product shelves inside the chamber. When it is entirely frozen, the chamber is evacuated and the partial vapour water pressure difference increases between the frozen ice zone and the chamber, thus creating a natural driving force for the sublimation process. Then, the heating system starts to provide enthalpy to accelerate the sublimation, which occurs at the moving ice front and proceeds from the top of the frozen product downwards. At the end of the primary stage of the process the major part of the water in the product has been removed and what remains is the portion of water constrained in the solution. At this point, the product can be unloaded, but the remaining water content is too high to be guaranteed its biological stability. For this purpose, to reduce the water content, the product is exposed at higher temperatures which induce the so called secondary drying stage of the freeze-drying process. In this study it is assumed only the primary drying phase of a small scale apparatus, for drying 50 vials filled with glycine in water adjusted to pH 3, with hydrochloric acid [17-18].

B. Initial conditions, quality criteria and constraints

Simulation experiments to control the heating shelves temperature, in notion to temperature inside the frozen product layer has been performed in Matlab & Simulink environments. According to chosen control loop, the system is nonlinear and non stationary one, since the product properties change over the drying cycle. *The following initial conditions for simulation experiments are assumed: $N_1=1$, $N_2=5$, $N_u=3$; System reference r=255K; Initial shelf temperature, before the start of the primary drying $Ts_{in}=228K$; Initial thickness of the interface front x=0.0023m; Thickness of the product L=0.003m.* During the primary drying the temperature of the heating shelves is maintained of about 298 K, for about 45 minutes of time, till the product is dried. The following constraints on the optimization problem are imposed: constraints on the amplitude of the control signal- the heating shelves temperature $228K< T_s < 298K$; constraints on the output changes- product temperature $238K< T_2 < 256K;$ constraints on the rate change of the control signal $0.5K< \Delta T_s < 3K.$

To estimate effectiveness of the drying process a parameter representing a notion between the cumulative energy which is minimized and the energy provided for the heating shelves, is introduced [12]. The notion will be the smallest possible, when the process is driven as fast as possible. The settling time of the process is also taken t_p. as quality criterion

C. *Experimental results*

The simulation study is realized using a validated plant model for a small scale lyophilization plant. The used model is derived from the physical laws of heat and mass transfer under minor assumptions [17-18]. To preserve the computational consistency, a final value of $x=0.0001\ m$ for the interface front is chosen to stop the simulation experiments. Increasing the heating shelves temperature from *228 K* indicates the start of the primary drying phase, confirmed by the initial drop of the product temperature, which represents the sudden loss of heat due to sublimation. The loss of heat due to sublimation vanishes after all of the unbound water has sublimed, then the enthalpy input from the shelf causes a sharp elevation of the product temperature.

Comparative experiments with the proposed predictive control strategies using NARX FN model with Gradient optimization procedure (case 1) and State-Space FN model with HQP (case 2), are made for two different values of the penalty term ρ and matrix R on the control actions over the control horizon. The temperature versus time profile for the product and heating shelves temperatures for the representative vial are presented on Fig. 2 for (case 1) and on Fig. 5 for (case 2). The decrease of the frozen layer is demonstrated on Fig. 3 for (case 1). The prediction of the states (case 2); x_1 - interface position (Fig. 7) and x_2 - temperature in the frozen region (Fig. 6), is presented. On Fig. 4 for (case 1) and Fig. 8 for (case 2) are demonstrated the squared errors of the models, during the controller operation.

The proposed FN models have a simplified structure based on the classical Takagi-Sugeno technique, which aims to ensure reliable and accurate modeling of the lyophilization process dynamics, stating small number of parameters without additional need of computational power. The consequent parameters of the proposed fuzzy-neural rules are initialized at first with randomly selected coefficients in a normalized range. The penalty terms/matrices into the objective functions are experimentally chosen. At each sampling period the models produce a predicted system output (the product temperature) in notion to current values of the input vector.

The optimization procedure for the NARX model lies on first-order gradient optimization algorithm, as iterative solution in notion to the computation of optimized values of the heating shelves temperature, by minimizing the system error for the temperature into the moving ice front. The presented case considers an unconstrained optimization problem where all process variables are scaled to their maximum allowed bounds. The control actions are computed analytically in an iterative manner along the defined control horizon. On the other hand, the optimization for the State-Space case is done by using a simple Hildreth Quadratic Programming procedure using the well known *Active Set* notation and a dual mechanism to determine the constraints that are not currently active. A system of inequality constraints is defined for the temperature of the heating shelves and its rate change, as well as for the product temperature.

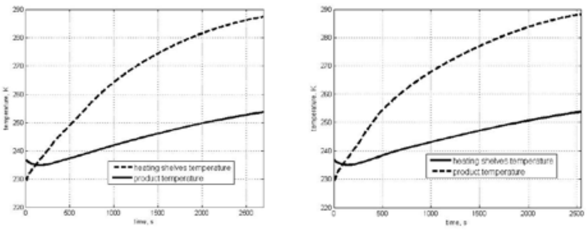

Figure 2. Temperature profile during lyophilization cycle (case 1a, b).

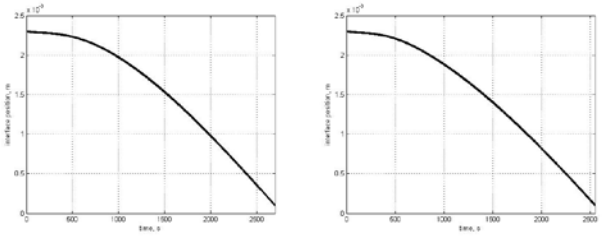

Figure 3. Interface position during lyophilization cycle (case 1a, b).

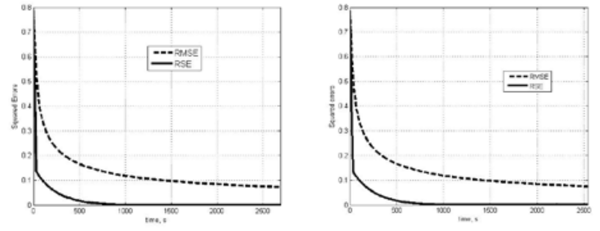

Figure 4. Squared model errors during lyophilization cycle (case 1a, b).

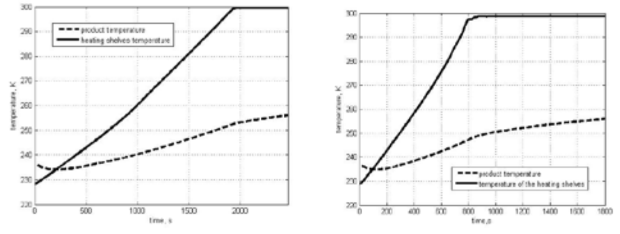

Figure 5. Temperature profile during lyophilization cycle (case 2a, b)

As can be seen for both cases the transient responses of the considered squared errors had a smooth nature and they are successfully minimized during the learning process for the models. This circumstance proves their proper operation and ensures a well driven lyophilization cycle demonstrated by the transient responses of the moving ice front. The moving ice front is an important parameter which accounts for the reliable an optimal drying process. For this purpose, it is used into the state-space model for a parameter being predicted by the model, along with the temperature in the frozen region. The assumed variations in case 1 differ on the chosen penalty term. As can be seen from Table 1, the increase of the penalty factor leads to improved process dynamics, intensification of the drying process and diminishing of the drying time. Similarly, the same variations are investigated in case 2 using two different *R* matrices. The obtained results are similar to case 1, except that the relation between the matrix *R* and the quality of process dynamics is reciprocal since; the notation in HQP optimization problem requires an indirect inversion of the Hessian.

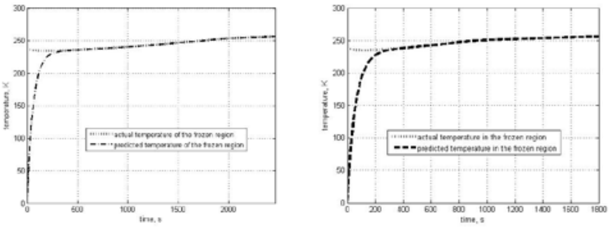

Figure 6. State x_2 (frozen region temperature) prediction (case 2a, b).

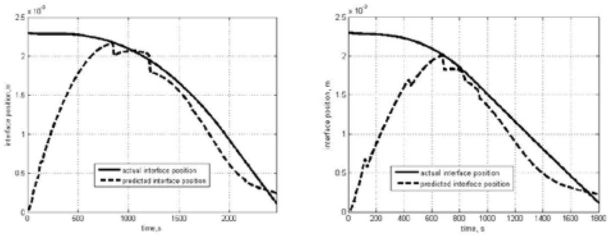

Figure 7. State x_1 (interface position) prediction (case 2a, b).

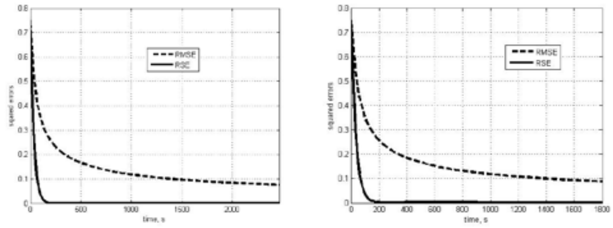

Figure 8. Squared model errors during lyophilization cycle (case 2a,b)

TABLE I. QUALITY CONTROL CRITERIA

case	ρ/R	t_p, s	RMSE	E_{ef}	T_s, K
1a	0.300	2649	0.072	0.029	287.3
1b	0.340	2549	0.074	0.028	288.2
2a	0.020	2460	0.076	0.025	298.3
2b	0.008	1830	0.088	0.023	298.4

Both control methodologies show a better system performance after proper selection of the initial conditions according to the requirements of the drying regime, but their real time applicability may impose different constraints depending on the product being lyophilized and the scale of the plant. The State Space approach will be a promising solution for large scale lyophilization plants, where the handling of the regime constraints is crucial and the system dynamic is relatively slow, which can accommodate with the computational procedures of the algorithm. On the other hand, the non-explicit strategy offers a simple and faster control solution, but unexpected disturbances may violate the scaling of the main parameters and deteriorate the system performance, which may restrict its application to small scale drying plants, where hard constraints on the regime are not imposed. A major advantage of the proposed control methodologies is the application of simple FN approach, which may impact the proper handling of some process uncertainties.

REFERENCES

[1] S. Patel, T. Doen and M. Pikal, “Determination of End Point of Primary Drying in Freeze-Drying Process Control,” AAPS PharmSciTech., vol. 11, no 1, pp. 73–84, March 2010.

[2] Y. Li and H. Kashiwa,”High-order Volterra Model Predictive Control and its application to a nonlinear polymerization process,” Int. J. of Autom. and Computing, vol. 2, no. 2, pp. 208-214, December 2005.

[3] R. Pisano, D. Fissore and A. Barresi, “Freeze-Drying Cycle Optimization Using Model Predictive Control Techniques,” Industrial Eng. and Chem. Res., vol. 50, no 12, pp. 7363–7379, April ,2011.

[4] R. Pisano, D. Fissore, D. Velardi and A. Barresi, “In-line optimization and control of an industrial freeze-drying process for pharmaceuticals,” Journal of Pharmaceutical Sciences, vol. 99, no 11, pp 4691-4709, April 2010.

[5] N. Daraoui, P. Dufour, H. Hammouri and A. Hottot, “Model predictive control during the primary drying stage of lyophilisation,” Control Engineering Practice, vol. 18, no 5, pp. 483-494, May 2010.

[6] P. Dufour, “Optimal operation of sublimation time of the freeze drying process by predictive control: Application of the MPC@CB Software,” Proc. of the 18th EFCE Symp., vol. 25, pp. 453-458, 2008.

[7] M. Pottman and R. Pearson, “Block-oriented NARMAX models with output multiplicities,”, AIChE, vol. 44, no. 1, pp. 131–140, 1998.

[8] J. Abonyi and R. Babuska, “Identification and control of nonlinear systems using fuzzy Hammerstein models,” Industrial and Engineering Chemistry Research, vol. 39, pp. 4302–4314, 2000.

[9] J. Gomez and E. Baeyens, “Hammerstein and Wiener model identification using rational orthogonal bases,” Latin American Applied Research, vol. 33, no. 3, 2003.

[10] R. Pearson, “Nonlinear input/output modeling,” Journal of Process Control, vol.5, no. 4, pp. 197-211, 1995.

[11] P. Stoica, “On the convergence of an iterative algorithm used for Hammerstein system identification,” IEEE Transactions on Automatic Control, vol. 26, pp. 967–969, 1981.

[12] Y. Todorov and M. Petrov, “Model predictive control of a Lyophlization plant: A simplified approach using Wiener and Hammerstein systems.” Journal of Control and Intelligent Systems, ACTA Press, vol. 39, no. 1, pp 23-32, January 2011.

[13] V. Chitanov and M. Petrov, “Multivariable Fuzzy-Neural Model of Polymer Process,” The Berkeley Electronic Press, Chemical Product and Process Modeling, vol. 4, no. 1, article 10, 2009.

[14] N. Shakev, A. Topalov and O. Kaynak, “A Neuro-Fuzzy Adaptive Sliding Mode Controller: Application to Second-Order Chaotic System.” Proc. of IEEE Conf. Intelligent Systems, pp. 9./14-19, 2008.

[15] J. Maciejowski, Predictive Control with constraints, Pearson, 2002.

[16] L. Wang, Model Predictive Control System Design and Implementation Using MATLAB, Springer, Verlag, 2009.

[17] M. Shoen and R. Jefferis III,”Simulation of a controlled freeze-drying process,” Int. J. of Mod. and Sim., vol. 20, no. 3, pp. 255-263, 2000.

[18] M. Schoen, B. Braxton, L. Gatlin and R. Jefferis III, “A Simulation Model for the Primary Drying Phase of the Freeze-Drying Cycle,” International Journal of Pharmaceutics, vol. 114, pp. 159-170, 1995.