MODEL PREDICTIVE CONTROL OF A LYOPHILIZATION PLANT: A NEWTON METHOD APPROACH

Y. Todorov*, S. Ahmed, M. Petrov****

**Institute of Cryobiology and Food Technologies, 53, Cherni Vrah blvd., 1407 Sofia, Bulgaria.*
***Technical University Sofia, branch Plovdiv, 25, Tsanko Dustabanov str., 4000, Plovdiv, Bulgaria*
e-mail: yancho.todorov@ieee.org, sevil.ahmed@gmail.com, mpetrov@tu-plovdiv.bg

Abstract: This paper describes a method for designing a nonlinear model predictive controller to be used in a lyophilization plant. The controller is based on a truncated fuzzy-neural Volterra predictive model and a simplified Newton method as optimization algorithm. The proposed approach is studied to control the product temperature in a lyophilization plant. The efficiency of the proposed approach is tested and proved by simulation experiments.

Key words: Fuzzy-Neural Models, Newton method, Predictive control, Lyophilization

INTRODUCTION

Lyophilization is a drying process in which the solvent and/or the suspension medium is crystallized at low temperatures and thereafter sublimed from the solid state directly into the vapour phase. Freeze-drying is mostly done with water as solvent. From the phase diagram of water can be seen the area in which this transfer from solid to vapour is possible. The drying transforms the ice or water in an amorphous phase into vapour. The goal of lyophilization is to produce a substance with good shelf stability and which is unchanged after reconstitution [1].

On the other hand, the lyophilized products are very expensive due to the high energy demands to maintain vacuum and refrigeration processes and also the latent heat for sublimation, as well. For this purpose is needed to be used an improved control strategies based on intelligent control methods, such as Model Predictive Control.

Model Predictive Control (MPC) has received a strong position when it comes to industrially implemented advanced control methodologies [2-3]. The main reason for this is the intuitive way MPC incorporates the process model in the controller design. In many problems relevant to the process control field today, the plant under control shows a strongly nonlinear behaviour. As a means to handle this, Nonlinear MPC (NMPC) is an often used method. NMPC, simply put, is model predictive control, where a nonlinear process model is used for prediction purposes, as opposed to a linear model for basic MPC [4-6].

Major industrial processes, are often nonlinear and the system nonlinearity cannot be ignored in practice. This has stimulated work in synthesizing MPC for use with a nonlinear analytical Volterra model and in Volterra series modelling. The main criticism in using Volterra series as nonlinear models lies in its large number of parameters needed to represent the kernels [7]. For this reason, in most practical solutions are imposed some structural restrictions to Volterra type models in order to attend a better model accuracy using a small number of parameters and to facilitate the identification procedures in notion to the computational effort. It has been shown, that any time-invariant nonlinear system can be approximated by a finite Volterra series to an arbitrary precision. Volterra models have the property to be linear in their parameters, i.e. the coefficients of their kernels, so that standard parameter estimation methods can be used [8].

In this paper, the proposed Volterra Fuzzy-Neural (VFN) model is implemented in MPC control scheme by using a simple fuzzy-neural approach and its efficiency is proved by simulation experiments in Matlab & Simulink environments to control the product temperature in a lyophilization plant.

DESING OF FUZZY-NEURAL VOLTERRA MODEL

Volterra models are widely used to model nonlinear processes. Since, with the increasing level of model nonlinearity, the number of its parameters increases sharply, in practice are mostly used truncated Volterra models [9]. In this approach, is considered the fuzzy-neural implementation of a second order Volterra model. As it is well known a wide class of nonlinear dynamic systems can be described in discrete time by NARX (Nonlinear AutoregRessive model with eXogenous inputs) input-output model. The used model in this paper is also taken in NARX type:

$$y(k)=f_y\,(x(k)) \quad \text{and}$$
$$x(k)=[u(k\text{-}1),\ldots,u(k\text{-}n_u),y(k\text{-}1),\ldots,y(k\text{-}n_y)] \tag{1}$$

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

$$R^{(i)}\text{:if } x_1 \text{ is } \tilde{A}_1^{(i)} \text{ and } x_p \text{ is } \tilde{A}_p^{(i)} \text{ then } f_y^{(i)}(k) \tag{2}$$

$$\begin{aligned} f_y^{(i)}(k)=&a_1^{(i)}y(k\text{-}1)+a_2^{(i)}y(k\text{-}2)+\ldots+a_{ny}^{(i)}y(k\text{-}n_y)+\\ &+b_1^{(i)}u(k\text{-}1)+b_2^{(i)}u(k\text{-}2)+\ldots+b_{nu}^{(i)}u(k\text{-}n_u)+b_o^{(i)}+\\ &+v_{m1}^{(i)}(k)+v_{m2}^{(i)}(k) \end{aligned} \tag{3}$$

where the Volterra kernels are:

$$v_{mi}=\sum_{i=1}^{n_u}\sum_{j=1}^{i}c_{ij}u(k\text{-}i)u(k\text{-}j) \tag{4}$$

The upper index $(i)=1,2,\ldots,N$ represents the number of the fuzzy rules, $\tilde{A}_i$ is an activated Gaussian fuzzy set defined in the universe of discourse of the input vector $\boldsymbol{x}$, the crisp

coefficients a_1, a_2,...,a_{ny}, b_1, b_2,...,b_{nu}, $c_{1,1}$, $c_{2,1}$, $c_{i,j}$ are the coefficients into the Sugeno function f_y and n_y /n_u is the history dependence on the input/output.
Finally, the actual implementations of the relevant fuzzy predictions have been obtained by appropriately shifting the inputs of the model. Therefore, a sequential algorithm based on the knowledge of current values of the regression vector, along with the fuzzy inference, computes:

$$\hat{y}(k+j) = \sum_{i=1}^{N} f_y^{(i)}(k+j)\overline{\mu}_y^{(i)}(k+j) \quad (5)$$

Fuzzy-neural model identification. The identification procedure involves structure identification of the process and estimation of the unknown parameters. The structure of the neuro-fuzzy model depends on the number of membership functions, their shape and the coefficients into the functions f_y in the consequent part of the rules (3). The task of model identification is to determine both groups of parameters of the Gaussian membership functions in the rule premise part and the linear parameters in the rule consequent part of the local models. A simplified fuzzy-neural approach is applied in this work, because of its simplicity and recurrent implementation of a tuning procedure for on-line applications [9].

The learning algorithm for the fuzzy-neural model is based on minimization of an instant error measurement function between the real plant output and the process output, calculated by the fuzzy-neural model:

$$E(k) = (y(k) - \hat{y}(k))^2 / 2 \quad (6)$$

where *y(k)* denotes the measured real plant output and $\hat{y}(k)$ is calculated by the fuzzy-neural network. The algorithm performs two steps gradient learning procedure. Assuming that β_{ij} is an adjustable i^{th} coefficient for the Sugeno function f_y into the j^{th} activated rule (2) as a connection in the output neuron, the general parameter learning rule for the consequent parameters is:

$$\beta_{ij}(k+1) = \beta_{ij}(k) + \eta\left(-\partial E / \partial \beta_{ij}\right) \quad (7)$$

After calculating the partial derivatives, the final recurrent predictions for each adjustable coefficient β_{ij} ($a^{(i)}$, $b^{(i)}$ or $c^{(i)}$) and the free coefficient are obtained by the following equations:

$$\beta_{ij}(k+1) = \beta_{ij}(k) + \eta\varepsilon(k)\overline{\mu}^{(i)} x_i(k) \quad (8)$$

$$\beta_{0j}(k+1) = \beta_{0j}(k) + \eta\varepsilon(k)\overline{\mu}^{(i)} x_i(k) \quad (9)$$

The output error *E* can be used back directly to the input layer, where there are the premise (center- Ωij and the deviation- σ_{ij} of a Gaussian fuzzy set) adjustable parameters. The error *E* is propagated through the links composed by the corresponded membership degrees, where the link weights are unit. Hence, the learning rule for the second group adjustable parameters in the input layer can be done by the same learning rule:

$$\Omega_{ij}(k+1) = \Omega_{ij}(k) + \eta(y-\hat{y})\overline{\mu}_y^{(i)}(k)[f_y^{(i)} - \hat{y}(k)]\frac{[x_i(k) - \Omega_{ij}(k)]}{\Omega_{ij}^2(k)} \quad (10)$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) + \eta(y-\hat{y})\overline{\mu}_y^{(i)}(k)[f_y^{(i)} - \hat{y}(k)]\frac{[x_i(k) - \sigma_{ij}(k)]^2}{\sigma_{ij}^3(k)} \quad (11)$$

BASICS OF THE APPLIED MODEL PREDICTIVE CONTROL STRATEGY

In recent years model predictive control has received a lot of attention in the control theory and applications. A model of the controlled process provides the forecast of the process output signal and the control signal is calculated in every step in a way that the difference between the reference and the output signal is minimized. The good system performance depends on model accuracy and parameters in the objective function. NMPC as it was applied with the VFN process model can be described in general with a block diagram, as it is depicted in (Fig. 1).

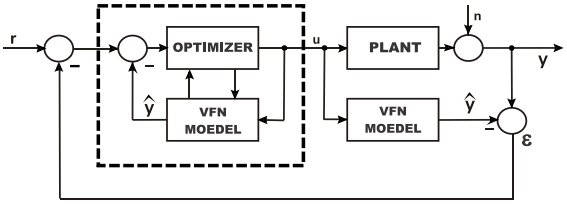

Fig.1. Block diagram of the proposed MPC system.

The selection of a minimization algorithm is a crucial issue in MPC, since this feature affects the computational efficiency of the control loop. Using the Newton method as optimization algorithm reduces the iterations to convergence in contrast to other techniques. The main cost of the Newton algorithm is the calculation of the Hessian matrix, but even with this overhead the low iteration numbers make the Newton faster algorithm for real time control [11]. As is well known, the Newton method is based on a quadratic approximation of an objective function as described:

$$\tilde{P}(x) = P(x^{(k)}) + \nabla^T P(x^{(k)})\Delta x^{(k)} + \frac{1}{2}(x^{(k)})^T \nabla^2 P(x^{(k)})\Delta x^{(k)} \quad (12)$$

This requires the evaluation of the Hessian and the gradient of the objective function. To implement the Newton method as an optimization algorithm is used the following recurrent equation:

$$x^{(k+1)} = x^{(k)} - H^{-1}(x^{(k)})\nabla P(x^{(k)}) \quad (13)$$

where *H* is the Hessian matrix with the second order partial derivatives as elements. An important principle in the Newton method is that the cost function must be quadratic one and the Hessian matrix must be positive definite.

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

$$J(k,u(k)) = \sum_{i=N_1}^{N_2}(r(k+i) - \hat{y}(k+i))^2 + \rho\sum_{i=N_1}^{Nu}\Delta u(k+i-1)^2 \quad (14)$$

where $\hat{y}$ is the predicted model output, *r* is the reference and *u* is the control action. The tuning parameters of the predictive controller are: N_1, N_2, N_u and ρ. N_1 is the minimum prediction horizon, N_2 is the maximum prediction horizon, N_u is the control horizon and ρ is the weighting factor penalizing changes in the control actions. When the criterion function is a quadratic one and there are no constraints on the control action, as well the cost function can be minimized analytically. If the criterion *J* is minimized with respect to the future control actions, then their optimal values can be calculated by applying the condition $\nabla J(k,U(k)) = 0$, where each element of the gradient vector can be calculated using the following equation:

$$\frac{\partial J(k,U(k))}{\partial u(k)} = [-2[R(k) - \hat{Y}(k)]^T \frac{\partial \hat{Y}(k)}{\partial U(k)} + 2\rho U(k)^T \frac{\partial \hat{U}(k))}{\partial U(k)}] \quad (15)$$

where *R(k)* is the system reference vector, $\hat{Y}(k)$ is the vector of the predicted model output and *U(k)* is the vector of the control actions.

Since, the VFN model consists a set of local sub models an explicit analytic solution of the above optimization problem

can be obtained. A simplified method for calculation of the elements of $\nabla J(k,U(k))$ based on the VFN model, is proposed here. Hence, according to f_y function (4) the unknown elements in (15) can be evaluated as follow:

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)} = \sum_{i=1}^{N} \left[\begin{array}{l} a_1{}^{(i)} \dfrac{\partial \hat{y}(k)}{\partial u(k)} + b_1{}^{(i)} \dfrac{\partial u(k)}{\partial u(k)} + \\ + c_{1,1}{}^{(i)} \dfrac{\partial u^2(k)}{\partial u(k)} + \\ + c_{2,1}{}^{(i)} \dfrac{\partial u(k-1)u(k)}{\partial u(k)} \end{array} \right] \overline{\mu}_y{}^{(i)}(k+1) \qquad (16)$$

$$\frac{\partial \hat{y}(k+N_2)}{\partial u(k)} = \sum_{i=1}^{N} \left[\begin{array}{l} a_1{}^{(i)} \dfrac{\partial \hat{y}(k+N_2-1)}{\partial u(k)} + \\ + a_2{}^{(i)} \dfrac{\partial \hat{y}(k+N_2-2)}{\partial u(k)} \end{array} \right] \overline{\mu}_y(k+N_2) \qquad (17)$$

Since, $\Delta u(k) = u(k) - u(k-1)$ then $\partial \hat{U} / \partial U$ represents a matrix with zeroes and ones. As Newton method imposes the implementation of the second order derivative of the cost function we can rewrite:

$$\nabla^2 J(k,U(k)) = \left[\frac{\partial^2 J(k,U(k))}{\partial u^2(k)}, ..., \frac{\partial^2 J(k,U(k))}{\partial u^2(k+N_u-1)} \right] \qquad (18)$$

$$\frac{\partial^2 J(k,U(k))}{\partial u^2(k)} = \left[\begin{array}{l} 2\left(\dfrac{\partial \hat{Y}k)}{\partial U(k)} \right)^2 - 2[R(k) - Y(k)] \dfrac{\partial^2 \hat{Y}(k)}{\partial U^2(k)} + \\ 2\rho \left(\dfrac{\partial \hat{U}(k))}{\partial U(k)} \right)^2 + 2\rho \left(U(k)^T \right) \dfrac{\partial^2 \hat{U}(k))}{\partial U^2(k)} \end{array} \right] \qquad (19)$$

Since $\partial^2 \hat{U}(k) / \partial U^2(k)$ always evaluates to zero, the second order derivatives $\partial^2 \hat{Y}(k) / \partial U^2(k)$ must be calculated, starting from the above mentioned equations. The Newton algorithm then iterates using the following expression:

$$-\frac{\partial J(k,U(k))}{\partial U(k)} = \frac{\partial^2 J(k,U(k))}{\partial U^2(k)} \Delta U(k) \qquad (20)$$

SIMULATION EXPERIMENTS

Experimental plant description. During the last years, extensive efforts by industry and research have been made to predict and optimize the course of the lyophilization cycles in order to control the quality of the product and to minimize the costs [11]. On (Fig.2) a simplified diagram of the main components of a lyophilization plant is shown. The plant consists particularly 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 and freeze or to supply heat to the product. This is supported by the shelves heater and refrigeration system (5). On those shelves the product is placed (6). The chamber is isolated from the condenser by the valve (7). The vacuum system is placed after the condenser. When the product is entirely frozen, the chamber is evacuated in order to increase the partial vapour water pressure difference between the frozen ice zone and the chamber. The shelf heating system starts to provide enthalpy for the sublimation process. The sublimation takes place at a moving ice front, which proceeds from the top of the frozen material downwards. The stage in which the remaining water content is further reduced is called secondary drying, which takes place at higher temperature. In this contribution is assumed only the first stage of the drying process called primary drying.

The considered plant is a small scale lyophilization apparatus, for drying of 50 vials filled with glycine in water adjusted to pH 3, with hydrochloric acid. The schematic diagram on (Fig. 2) depicts the sublimation process occurring at the interface which is located at a distance *x* from the vial bottom. During sublimation the interface moves in a negative direction, while the product height remains constant.

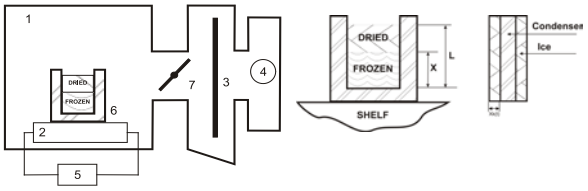

Fig.2. Schematic diagram of a simplified Lyophilization plant.

Simulation experiments. Simulation experiments in *Matlab* & *Simulink* environments to control the heating shelves temperature, in notion to temperature inside the frozen product layer, are made. According to this circumstance, the system is nonlinear and non stationary one and this is because during the sublimation process the properties of the product are changed. The following initial conditions for simulation experiments are assumed; $N_1=1$, $N_2=3$, $N_u=3$, system reference $r=255$ *K*, initial shelf temperature, $Ts_{in} = 228$ *K*, initial thickness of the front $x=0.0023$ *m*, thickness of the product $L=0.003$ *m*. In the primary drying stage it is required to maintain the shelf temperature about *298 K*, until the product is dried. This circumstance requires about *45* minutes of time for the primary drying stage of the process.

The aim of the control system is to reduce the system error between the reference product temperature and the current product temperature at each sampling period, by calculating an appropriate control action, which will drive the drying process as fast as possible. The physical explanation of this is minimizing the energy for the drying process, as computing the optimized values for the heating shelves temperatures. According to this is defined a criterion in which the efficiency parameter E_{ef} represents a notion between the cumulative energy which is minimized and the energy provided for the heating process [10].

$$E_{ef} = \left(\int_0^t (T_r(t) - T_p(t))dt \right) / \left(\int_0^t T_s(t)dt \right) \qquad (21)$$

where: *Ts – temperature of the heating shelves, Tr – reference product temperature, Tp – product temperature.* As a reference criterion for the process is also taken the settling time of the process (t_p). Evaluation of the model performance is demonstrated by *Root Mean Squared Error* (RMSE) and *Root Squared Error* (RSE) plots of the model.

There were made comparative experiments with the proposed VFN model using the Newton method as optimization algorithm and the standard Gradient optimization algorithm, as reference. The validated plant model used as plant process for simulation in this study was derived from the physical laws of heat and mass transfer for a typical laboratory plant. The temperature versus time profile for the product and heating shelf temperatures for the representative vial is presented on (Fig.3) and (Fig.6). The primary drying phase for the cycle was started by increasing the shelf temperature from *228 K*. The initial drop of the product temperature represents the sudden loss of heat due to sublimation and indicates the start of the primary drying. After, all of the unbound water has sublimated, the loss of heat due to sublimation vanishes and the enthalpy input from the shelf causes a sharp elevation of the product temperature. The VFN model responses of the RMSE and RSE are shown on (Fig.5) and (Fig.8). On (Fig.4)

and (Fig.7) is demonstrated the decrease of the frozen layer interface *x.*

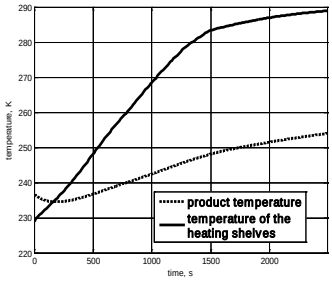

Fig.3. Product and shelf temperatures using the Newton method.

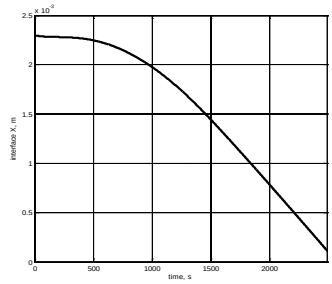

Fig.4. Interface position using the Newton method.

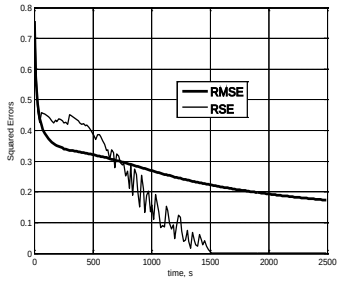

Fig.5. RMSE and RSE responses of the model using the Newton method.

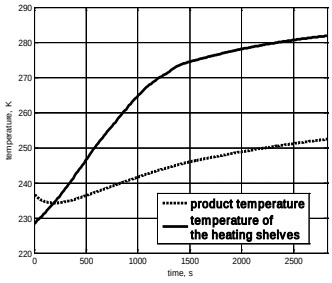

Fig.6. Product and shelf temperatures using the Gradient method.

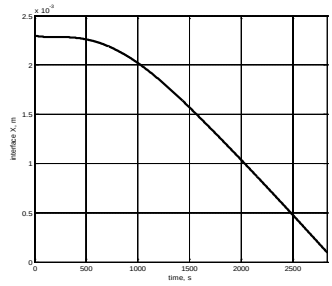

Fig.7. Interface position using the Gradient method..

Using the proposed Newton algorithm and the same initial conditions we can decrease the drying time in contrast to the classical Gradient optimization method. The use of the proposed control strategy is also an effective energy saving solution for the process, since the maximum allowed shelf temperature is under its maximum bound.

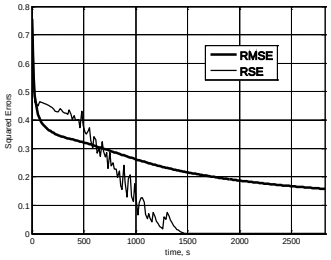

Fig.8. RMSE and RSE responses of the model using Gradient method.

Table.1

Method	t_p	E_{ef}	RMSE
Gradient	2828	0.000660	0.1735
Newton	2490	0.000757	0.1575

CONCLUSIONS

It was presented in this paper a method for designing a nonlinear MPC. The controller is based on a truncated VFN model and Newton method as optimization algorithm. The proposed approach was used to control the product temperature in a lyophilization plant. The simulation experiments show the efficiency of the proposed control strategy. The product temperature in the frozen region rises according to lyophilization cycle regime requirements and constraints and using the Newton method as optimization procedure reduces the drying time in contrast to classical Gradient descent procedure.

ACKNOWLEDGMENTS

The authors acknowledge the Ministry of Education, Youth and Science, Bulgaria, Research Project BY-TH-108/2005.

REFERENCES

1. G. Oetjen, P. Hasley, Freeze-Drying 2-nded., WILEY, 2004.
2. J. Qin, T. Bagwell, A survey of industrial model predictive control technology, Cont. Eng. Pract., 11(7), 733-764, 2003.
3. H. Takatsu, Advanced process control and optimization solutions, Yokogava Tech. Report. Ed., 43, 2007.
4. E. Camacho, C. Bordons, Model Predictive Control 2nded., Springer Verlag, 2004.
5. B. Rawlings, Tutorial overview of model predictive control, IEEE Contr. Syst. Magazine, 20(3): 38–52, 2000.
6. F. Allgöwer, R. Findeisen, An introduction to nonlinear model predictive control, In. Proc. of 21st Benelux Meeting on Systems and Control, Veldhoven, 2003.
7. G. Favier, A. Kibangou, A. Khouaja, Nonlinear system modeling by means of Volterra models: Approaches for parametric complexity reduction, Tech. Sym. on Innovative Strategies in Modeling and Robust Control of Ind. Proc., 2004.
8. Q. Zheng, E. Zafirou, Nonlinear system identification using Volterra-Laguerre expansion, ACC, Vol.3, 2195-2199,1995.
9. S. Kozak, P. Taraba, MPC using AR Volterra models, The 11th Med. Conf. on Control and Automation, t5-002, 2003.
10. Y. Todorov, Ts. Tsvetkov, Volterra model predictive control of a Lyophilization plant. Int. IEEE conference Intelligent systems, Bulgaria, Vol.3, 20/13-18, 2008.
11. D. Soloway, P. Hasley, Neural Generalized Predictive Control: A Newton-Raphson Implementation. NASA Technical memorandum 110244, 1997.
12. M. Shoen, A simulation model for primary drying phase of freeze-drying. Int. Journal of Pharm., Vol.114, 159-170, 1995.