Fuzzy-Neural Predictive Control using Levenberg-Marquardt optimization approach

Yancho Todorov, *IEEE Member*
Department of "Intelligent Systems",
Institute of Information and Communication
Technologies, Bulgarian Academy of Sciences,
Sofia, Bulgaria
yancho.todorov@ieee.org

Margarita Terzyiska, Sevil Ahmed, Michail Petrov
Department of "Control Systems",
Technical University-Sofia, branch Plovdiv,
Plovdiv, Bulgaria
terzyiska@gmail.com
[sevil.ahmed, mpetrov]@tu-plovdiv.bg

***Abstract*—It is proposed in this paper a study on the influence of the *Levenberg-Marquardt* optimization approach for computation of the control actions in Nonlinear Model Predictive Controller. To predict the future plant behavior, a classical Takagi-Sugeno inference is used. A comparison by applying the *Gradient descent* and the *Newton-Raphson* optimization approaches is made. The efficiency of the proposed optimization strategies is demonstrated by experiments in *MATLAB* environment to control a Continuous Stirred Tank Reactor.**

Keywords—Takagi-Sugeno model, Nonlinear Predictive Control, Optimization, Gradient descent, Newton-Raphson, Levenberg- Marcquart

I. INTRODUCTION

Model Predictive Control (MPC) algorithms based on linear models have been successfully used for years in numerous industrial applications [1]. In [2] is provided a review of the most commonly used methods that have been embedded in an industrial model predictive control. Since, the nature of most industrial processes is inherently nonlinear this implies the use of nonlinear models and respectively Nonlinear Model Predictive Control (NMPC) algorithms.

NMPC is a variant of MPC that is characterized by the use of nonlinear system models in the prediction stage. As in linear MPC, NMPC requires the iterative solution of optimal control problems on a finite prediction horizon. While these problems are convex in linear MPC, in nonlinear MPC they are not convex anymore. The nonlinear optimization task must be solved online. Therefore, the success of a NMPC algorithm depends critically on the used model. For that purpose, it is very important to find a predictive model that effectively describes the nonlinear behavior of the system and can easily be incorporated into NMPC algorithm.

One possibility is to use first principle models such as nonlinear ordinary differential equations, partial differential equations, integro-differential equations and delay equations models. Such models can be accurate over a wide range of operating conditions, but they are difficult to develop for many industrial cases and may lead to numerical problems (e.g. stiffness, ill-conditioning). The other possibility is to use empirical or black–box models (e.g. neural networks, fuzzy models, polynomial models, Volterra series models) [3-10]. How to select a suitable nonlinear model for NMPC is described in details in [11].

The fusion of the fuzzy logic with the neural networks allows to combine the learning and computational ability of neural networks with the human like IF-THEN reasoning of a fuzzy system. This could be compared with the human brain [12-13] – neural network concentrate on the structure of human brain, i.e., on the "hardware" whereas fuzzy logic system concentrate on "software". Combining neural networks and fuzzy systems in one unified framework has become popular in the last few years. Also, any methods have been proposed in the literatures that combine fuzzy-neural network and model predictive control algorithms [14-16]. In recent years, a general approach based on multiple LTI models around various function points has been proposed. The so-called multiple models, Takagi-Sugeno (TS) approach is a convex polytopic representation, which can be obtained either through mathematical transformation or through achieved linearization around various operating points [17-19].

The optimization procedure takes an important role in MPC, computing an optimal trajectory of future controls in a way to be optimised the future process behaviour. Since, the control problem is formulated as nonlinear optimization task in real manufacturing, analytical approaches cannot be appropriately employed. Hence, it is usually solved by a means of iterative optimization algorithms. For this purpose, different numerical approaches have been proposed in literature, as suggested in [20] the Newton-Raphson to solve the nonlinear optimization and the use of Gradient descent [21]. Also, in [22-23] are used sequential quadratic programming and Levenberg-Marquardt (LM).

The presented work deals with the development of a fuzzy-neural model predictive controller based on classical Takagi-Sugeno inference as predictive model and the use of Gradient descent, Newton-Raphson and Levenberg-Marquardt optimization strategies to compute an optimal control trajectory. A comparison by simulation experiments in MATLAB environment, on the influence of the optimization policy in trajectory reference tracking during the process control of a Continuous Stirred Tank Reactor (CSTR) has been made.

978-1-4799-0661-1/13/$31.00 ©2013 IEEE

II. FUZZY-NEURAL PREDICTIVE MODEL

A. Takagi-Sugeno fuzzy-neural model

Since the middle of the 1980s, TS fuzzy models have attracted a great deal of attention from industrial practitioners and academic researchers, especially because they can effectively approximate a wide class of nonlinear systems [18]. Thus, in discrete time by using the NARX representation model (Nonlinear AutoregRessive model with eXogenous inputs) can be derived:

$$y(k) = f_y(x(k)) \tag{1}$$

where the unknown nonlinear function f_y can be approximated by Takagi-Sugeno type fuzzy rules:

$$\begin{aligned} &R^{(i)}: if\ x_1\ is\ \tilde{A}_1^{(i)}\ and\ x_p\ is\ \tilde{A}_p^{(i)}\ then\ f_y^{(i)}(k) \\ &where\ f_y^{(i)}(k) = a_1^{(i)} y(k-1) + a_2^{(i)} y(k-2) + \cdots + a_{ny}^{(i)} y(k-n_y) + \\ &+ b_1^{(i)} u(k) + b_2^{(i)} u(k-1) + \cdots + b_{nu}^{(i)} u(k-n_u) + b_0^{(i)} \end{aligned} \tag{2}$$

where $(i)=1,2,...N$ and denotes the number of the fuzzy rules. A_i is an activated fuzzy set defined in the universe of discourse of the input x_i and the crisp coefficients $a_1, a_2,...a_{ny}, b_1, b_2,...b_{nu}$ are the coefficients into the Sugeno function f_y.

The identification of a TS model requires the two main groups of unknown parameters to be determined: the number of membership functions, their shape and the parameters of the function f_y in the consequent part of the rules. A two step gradient learning procedure based on minimization of an instant error measurement function between the process output and the model output is implemented [7-10].

During the learning process, two groups of parameters in the fuzzy neural architecture – premise and consequent parameters are under adaptation. The consequent parameters are the coefficients $a_1, a_2,...a_{ny}, b_1, b_2,...b_{nu}$ in the Sugeno function f_y and they are calculated by the following equations:

$$\begin{aligned} \beta_{ij}(k+1) &= \beta_{ij}(k) + \eta(y - y_M)\bar{\mu}_y^{(j)}(k) x_i(k) \\ \beta_{0j}(k+1) &= \beta_{0j}(k) + \eta(y - y_M)\bar{\mu}_y^{(j)}(k) \end{aligned} \tag{3}$$

where η is the learning rate and β_{ij} is an adjustable i^{-th} coefficient (a_i or b_i) in the Sugeno function f_y of the j^{-th} activated rule.

The premise parameters are the centre c_{ij} and the deviation σ_{ij} of a Gaussian fuzzy set. Using the following equations, they are calculated as:

$$\begin{aligned} c_{ij}(k+1) &= c_{ij}(k) + \eta(y - y_M)\bar{\mu}_y^{(j)}(k)[f_y^{(i)} - \hat{y}(k)]\frac{[x_i(k) - c_{ij}(k)]}{c_{ij}^2(k)} \\ \sigma_{ij}(k+1) &= \sigma_{ij}(k) + \eta(y - y_M)\bar{\mu}_y^{(j)}(k)[f_y^{(i)} - \hat{y}(k)]\frac{[x_i(k) - \sigma_{ij}(k)]^3}{\sigma_{ij}^2(k)} \end{aligned} \tag{4}$$

III. MODEL PREDICTIVE CONTROL POLICY

Using the designed TS fuzzy-neural model, the *Optimization Algorithm* computes the future control actions at each sampling period, by minimizing the following cost function:

$$J(k, u(k)) = \sum_{i=N_1}^{N_2} (r(k+i) - \hat{y}(k+i))^2 + \rho \sum_{i=1}^{N_u} \Delta u(k+i-1)^2 \tag{5}$$

where $\hat{y}$ is the predicted model output, r is the reference signal and u is the control action. The tuning parameters of the stated predictive controller are: N_1, N_2, N_u and ρ. N_1 and N_2 are the minimum/maximum prediction horizons, N_u is the control horizon and ρ is the weighting factor penalizing changes in the control actions.

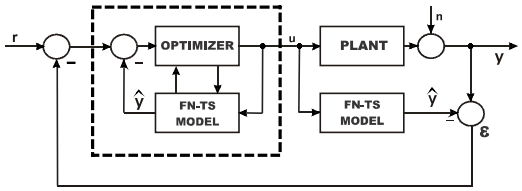

Fig. 1. Schematic diagram of the proposed control system

Since the criterion function is a quadratic one and there are no imposed constraints on the control action, the minimization procedure can be performed iteratively. If the criterion J is minimized with respect to the future control moves Δu, then their optimal values can be calculated by applying the condition $\nabla J[k,\ U(k)] = 0$.

A. Gradient Descent optimization approach

Each element from gradient vector is calculated using the following equation:

$$\frac{\partial J[k, U(k)]}{\partial U(k)} = \left[-2[R(k) - \hat{Y}(k)]^T \frac{\partial \hat{Y}(k)}{\partial U(k)} + 2\rho \hat{U}(k)^T \frac{\partial \hat{U}(k)}{\partial U(k)} \right] \tag{6}$$

It can be seen, that two partial derivatives have to be determined. The first one is $\partial \hat{Y}(k)/\partial U(k)$, and second one is $\partial \hat{U}(k)/\partial U(k^2$. Each element from the first group of partial derivatives is calculated by the following equations [7]:

$$\frac{\partial \hat{y}(k)}{\partial u(k)} = \sum_{i=1}^{N} b_1^{(i)} \bar{\mu}_y^{(i)}(k) \tag{7}$$

$$\frac{\partial \hat{y}(k + N_2)}{\partial u(k)} = \sum_{i=1}^{N} \left[\begin{matrix} a_1^{(i)} \dfrac{\partial \hat{y}(k + N_2 - 1)}{\partial u(k)} + \cdots \\ + a_2^{(i)} \dfrac{\partial \hat{y}(k + N_2 - 2)}{\partial u(k)} \end{matrix} \right] \bar{\mu}_y^{(i)}(k + N_2) \tag{8}$$

The second group partial derivatives $\partial \hat{U}(k)/\partial U(k)$ represents a matrix of size $N_2 \times N_u$. Since, $\Delta u(k) = u(k) - u(k-1)$ the elements of the matrix are zeros and ones. Thus, the recurrent equations for calculation of the control actions along the horizon are derived as:

$$\Delta u(k+N_u-1)= \rho^{-1}\left[e(k+N_1)\frac{\partial\hat{y}(k+N_1)}{\partial u(k+N_u-1)}+\ldots+e(k+N_2)\frac{\partial\hat{y}(k+N_2)}{\partial u(k+N_u-1)}\right] \quad (9)$$

$$\Delta u(k)=\Delta u(k+1)+ \rho^{-1}\left[e(k+N_1)\frac{\partial\hat{y}(k+N_1)}{\partial u(k)}+\ldots+e(k+N_2)\frac{\partial\hat{y}(k+N_2)}{\partial u(k)}\right] \quad (10)$$

B. Newton-Raphson optimization approach

In order to increase the efficiency of the gradient search, a class of optimization algorithms that realize quadratic approximation by using second order derivatives, are developed. The Newton-Raphson and Levenberg-Marquardt optimization algorithms belong to this class. The main cost of these two optimization strategies is the calculation of the Hessian matrix, but even with this overhead the low iteration numbers make the algorithms faster for real time control [20].

It was already mentioned, that the *Newton-Raphson* method, or *Newton* method, is based on a quadratic approximation of an objective function as follows:

$$\tilde{P}(x)=P(x^{(k)})+\nabla^T P(x^{(k)})\Delta x^{(k)}+\frac{1}{2}(x^{(k)})^T\nabla^2 P(x^{(k)})\Delta x^{(k)} \quad (11)$$

That requires the evaluation of the Hessian and the gradient of the objective function. To implement the Newton method as an optimization algorithm the following recurrent equations are used:

$$\begin{aligned} x^{(k+1)} &= x^{(k)}-\left[\nabla^2 P(x^{(k)})\right]^{-1}\nabla P(x^{(k)}) \\ x^{(k+1)} &= x^{(k)}-H^{-1}(x^{(k)})\nabla P(x^{(k)}) \end{aligned} \quad (12)$$

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. As imposed, the implementation of the second order derivatives of the cost function it can be rewritten [24]:

$$\frac{\partial^2 J(k,U(k))}{\partial u^2(k)}=\begin{bmatrix} 2\left(\frac{\partial\hat{Y}k)}{\partial U(k)}\right)^2-2[R(k)-Y(k)]\frac{\partial^2\hat{Y}(k)}{\partial U^2(k)}+ \\ 2\rho\left(\frac{\partial\hat{U}(k))}{\partial U(k)}\right)^2+2\rho\left(U(k)^T\right)\frac{\partial^2\hat{U}(k))}{\partial U^2(k)} \end{bmatrix} \quad (13)$$

Since $\partial^2\hat{U}(k)/\partial U(k)^2$ and $\partial^2\hat{Y}(k)/\partial U(k)^2$ always evaluate to zero, the Hessian has the following form:

$$H[k,U(k)]=-2\left(\frac{\partial\hat{Y}(k)}{\partial U(k)}\right)^2+2\rho\left(\frac{\partial\hat{U}(k)}{\partial U(k)}\right)^2 \quad (14)$$

Taking into account the above equations, the first order partial derivatives are calculated in the same manner. Then the algorithm iterates along the control horizon as:

$$\frac{\partial^2 J[k,U(k)]}{\partial U^2(k)}=\left[-2\left(\frac{\partial\hat{Y}(k)}{\partial U(k)}\right)^2+2\rho\left(\frac{\partial\hat{U}(k)}{\partial U(k)}\right)^2\right]^{-1} *\left[-2\left[R(k)-\hat{Y}(k)\right]^T\frac{\partial\hat{Y}(k)}{\partial U(k)}+2\rho\hat{U}(k)^T\frac{\partial\hat{U}(k)}{\partial U(k)}\right] \quad (15)$$

$$\Delta u(k+N_u-1)= \rho^{-1}H^{-1}\begin{bmatrix} e(k+N_1)\frac{\partial\hat{y}(k+N_1)}{\partial u(k+N_u-1)}+\ldots+ \\ +e(k+N_2)\frac{\partial\hat{y}(k+N_2)}{\partial u(k+N_u-1)} \end{bmatrix} \quad (16)$$

$$\Delta u(k)=\Delta u(k+1)+ \rho^{-1}H^{-1}\begin{bmatrix} e(k+N_1)\frac{\partial\hat{y}(k+N_1)}{\partial u(k)}+\ldots+ \\ +e(k+N_2)\frac{\partial\hat{y}(k+N_2)}{\partial u(k)} \end{bmatrix} \quad (17)$$

C. The Levenberg-Marquardt algorithm

The Levenberg-Marquardt algorithm also uses the approximated Hessian and the information in the gradient, taking into account some regularization factors. The algorithm iterates using the following general equation [23]:

$$x^{(k+1)}=x^{(k)}-[H^{-1}(x^{(k)})+\lambda E]\nabla Q(x^{(k)}) \quad (18)$$

where H is the Hessian as it is computed in (14), E is the identity matrix and λ is the Levenberg-Marquardt parameter, which adjust the direction of movement to extremes, from gradient method at great value ($\lambda > 10^3$) to Newton method (12) when $\lambda = 0$.

$$\Delta u(k+N_u-1)=\rho^{-1}[H^{-1}+\lambda E]* \begin{bmatrix} e(k+N_1)\frac{\partial\hat{y}(k+N_1)}{\partial u(k+N_u-1)}+\ldots \\ +e(k+N_2)\frac{\partial\hat{y}(k+N_2)}{\partial u(k+N_u-1)} \end{bmatrix} \quad (19)$$

$$\Delta u(k)=\Delta u(k+1)+\rho^{-1}[H^{-1}+\lambda E]* *\begin{bmatrix} e(k+N_1)\frac{\partial\hat{y}(k+N_1)}{\partial u(k)}+\ldots+ \\ +e(k+N_2)\frac{\partial\hat{y}(k+N_2)}{\partial u(k)} \end{bmatrix} \quad (20)$$

When λ_κ parameter is small, the method represents a quadratic approximation and when it is large, the Hessian is negligible and the LM method works similarly as Gradient descent algorithm. At first iterations, LM works as a gradient method and as it gets near the optimal point it gradually switches to Newton based approach. When LM parameter gets smaller, LM finds a locally linear solution, precisely and quickly. After each iteration of the search, Hessian is checked to be positive definite (convex optimization). If Hessian is not positive definite, λ_κ is increased until this happens. To investigate the positive definiteness of Hessian, Cholesky factorization has to be used [23]. To find the minimum using

LM it is necessary to calculate the Gradient and the Hessian of the cost function. Using a TS fuzzy-neural network model of the system, it is straightforward task.

IV. SIMULATION RESULTS

The plant model used for simulations is a simplified nonlinear model of a CSTR, whose dynamic equations are given by:

$$\begin{aligned} \dot{x}_1 &= -x_1 + D_a(1-x_1)\exp\left(\frac{x_2}{1+x_2/\phi}\right) \\ \dot{x}_2 &= -(1+\delta)x_2 + BD_a(1-x_1)\exp\left(\frac{x_2}{1+x_2/\phi}\right) + \delta u \end{aligned} \tag{21}$$

where x_1 and x_2 represent the dimensionless reactant concentration and the reactor temperature, respectively. The control action u is dimensionless cooling jacket temperature. The physical parameters in the CSTR model equations are D_a, φ, B and δ which correspond to the Damkhler number, the activated energy, the heat of reaction and the heat transfer coefficient, respectively. Based on the nominal values of the system parameters, D_a=0.072, φ=20, B=8 and δ=0.69, the open-loop CSTR exhibits three steady states $(x_1,x_2)_A$=(0.144, 0.886), $(x_1,x_2)_B$=(0.445, 2.75) and $(x_1,x_2)_C$=(0.765, 4.705), where the upper and the lower steady states are stable, whereas the middle one is unstable. The control objective here is to bring the nonlinear CSTR from the stable equilibrium point $(x_1,x_2)_A$ to the unstable one $(x_1,x_2)_B$ [25].

In the present study different control simulations in MATLAB environment are performed. At first, the used TS model for output predictions is initialized with random coefficients, in order to ensure the random directions for the beginning of the optimization step. These initials are used in all of the considered experiments to compare the performances of the selected optimization policies. The horizons are set as: N_1=1, N_2=5, N_u=3.

On Fig. 2 is demonstrated the performance of the LM algorithm in case of variable system reference and nonminimum phase behavior of the plant. After the 300th time step the change of the heath transfer coefficient δ in the second heat reaction, causes system deterioration. On Fig. 3 is shown the dynamic behavior of the λ_k during the control process. As can be seen in the start point the parameter has greater values of about 9000 which is smoothly decreased to values closer to 0. Recalling in mind (19), this result shows that at first time steps the LM algorithm works as gradient algorithm, faster approaching to the minimum. Afterwards, being closer to optimum, it starts to operate as Newton approach in order to ensure smoothly that the minimum has been reached.

The Fig. 4 demonstrates the influence of the selection of λ_k on the convergence and performance of the LM optimization approach. As can be seen, the greeter values of this parameter ensure a faster controller performance by diminishing the time for reaction in account of the slightly changed system overshoot. This can be crucial, especially in faster and sudden chemical reactions in chemical processes, as in the considered case.

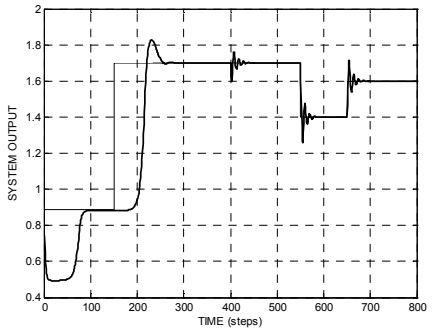

Fig. 2. Transient process response in case of variable system reference and plant parameter changes

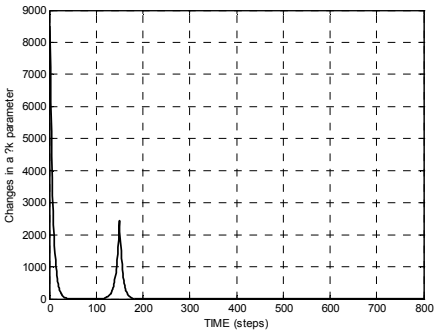

Fig. 3. Changes in a λ_K parameter

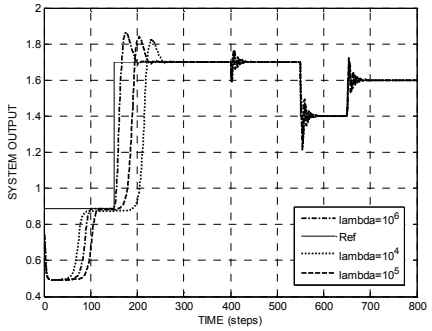

Fig. 4. Transient process response in case of variable system reference, plant parameter changes and different initail values of λ_k

On Fig. 5 is demonstrated the influence of the penalty factor ρ in the control term of the performance index on the system behavior. The obtained results show an expected behavior of the scaling influence of ρ on the Δu, which affects the settling time of the controlled variable and the system overshoot. The performance of the proposed three optimization polices is compared on Fig. 6. As can be seen from the experiments, using equal initial conditions for the model and the scaling factor of the controller, a different system performance is reached. The Newton and LM approaches as being of same family methods give a similar faster system performance, compared to Gradient Descent. This, in fact shows that the scaling factor value depends on the used optimization policy instead of the method of its selection. Comparing the Newton and LM methods, the latter ensures a

better system performance due to its nature to switch between other two approaches during the minimization procedure, which affects the settling time of the process and reduce the system overshoot. A less aggressive and more robust control policy is achieved.

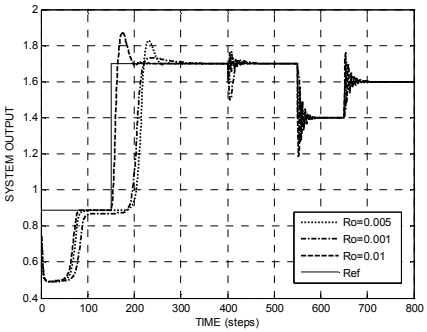

Fig. 5. Transient process response in case of variable system reference, plant parameter changes and different initial values of ρ

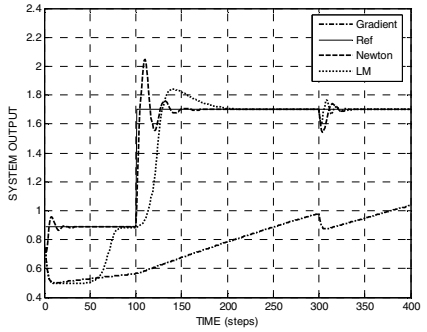

Fig. 6. Transient process response in case of variable system reference, plant parameter changes and different optimization policies

V. CONCLUSIONS

It was presented in this paper a FN predictive controller using LM optimization algorithm. It was shown that the NMPC with LM ensures a reliable system performance in case of variable system reference and plant parameter changes. The simulations demonstrate the influence of λ_k and ρ on the predictive control algorithm performance. The proposed NMPC with LM optimization is compared to nonlinear predictive controllers, that use Gradient Descent and NR optimization approaches. The proposed Newton like algorithms are rapidly converging when an analytical minimization approach is considered. Their main disadvantage is the need to obtain invert Hesse matrix during the algorithm iteration.

ACKNOWLEDGMENT

The research work reported in the paper is partly supported by the project ***AComIn*** "Advanced Computing for Innovation", grant 316087, funded by the FP7 Capacity Programme (Research Potential of Convergence Regions).

REFERENCES

[1] S. J Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," Control Eng. Pract., vol. 11, pp. 733–764, 2003.

[2] K. S. Holkar and L. M. Waghmare, "An Overview of Model Predictive Control," Int. Journal of Control and Automation vol. 3(4), 2010.

[3] J. Mendes and Araujo Rui, "Fuzzy Model Predictive Control for Nonlinear Processes," In Proc. of 17th IEEE International Conference on Emerging Technologies & Factory Automation, September, 2012.

[4] Li Zheng, "Intelligent Fuzzy Predictive Controller Design for Multivariable Process System," Journal of Computational Information Systems, vol. 6(9), pp. 3003-3011, 2010.

[5] Y. F. Zhang, Z. G Su and P.H Wang, "A convenient version of T-S fuzzy model with enhanced performance," In Proc. of the 8th international conference on Fuzzy Systems and Knowldege Discovery, pp. 1074-1079, 2011.

[6] J. Paulusová, and M. Dúbravská, "Hybrid Predictive Controller Based on Fuzzy-Neuro Model," In Proc. of 17th International Conference on Process Control, Slovak Republic, June 9 – 12, 2009.

[7] M. Terzyiska, Y. Todorov, A. Mitev and M. Petrov, "Nonlinear Model Based Predictive Controller using a Fuzzy-Neural Hammerstein model," In Proc. of the international conference 'Modern Trends in Control', Technical University of Kosice, Slovakia, pp. 299-308, 2006.

[8] Y. Todorov, S. Ahmed and M. Petrov, "Volterra Model Predictive Control of a Lyophilization plant: A Newton Optimization approach," Inf. Tech. and Control, Submitted, year IX, vol. 4, pp. 9-15, 2011.

[9] Y. Todorov and Ts. Tsvetkov, "Volterra Model Predictive Control of a Lyophilization plant," In Proceedings of the 4-th International IEEE conference "Intelligent Systems", vol. 3, pp. 20-13/20-18, 2008.

[10] S. Ahmed, "Fuzzy-Neural Model Predictive Control Of An Evaporator System," In Proc. of Int. Conf. Automatics and Informatics'11, Sofia, Bulgaria. pp. B-93–B-96, 2011

[11] R. Pearson, "Selecting nonlinear model structures for computer control," Journal of Process Control, vol. 13, 2003.

[12] Ch. R. Alavala, Fuzzy Logic and Neural Networks: Basic Concepts and Applications, New Age Int. Pvt. Ltd. Publishers, December 2008.

[13] K. Passino and S. Yourkovic, Fuzzy Control, Adisson-Wesley, 1998.

[14] L. Dalhoumi, "Fuzzy predictive control based on Takagi-Sugeno model for nonlinear systems," In Proc. of 7th International Multi-Conference on Systems Signals and Devices (SSD), pp. 1-7, 2010.

[15] M. Hadjili, "Modelling and control using Takagi-Sugeno fuzzy models," In Proc. of Electronics, Comm. and Photonics Conf. pp. 1-6, 2011.

[16] J. Mendes, "Adaptive fuzzy generalized predictive control based on Discrete-Time T-S fuzzy model," In Proc. IEEE Conference of Emerging Technologies and Factory Automation, pp. 1-8, 2010.

[17] M. Chadli and P. Borne, Multiple Models Approach in Automation: Takagi-Sugeno Fuzzy Systems, Wiley, 2012.

[18] V. Chitanov, C. Kiparissides and M. Petrov, "Neural - Fuzzy Modeling of Polymer Quality in Batch Polymerization Reactors," In Proceedings of the 2nd International IEEE Conference on "Intelligent Systems", vol. III, pp. 67 – 72, 2004.

[19] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," IEEE Transactions on Systems, Man, Cybernetics, vol. 15 (1), pp. 116–132, 1985.

[20] D.Soloway and P. J. Haley, "Neural GeneralizedPredictive Control; A Newton-Raphson Implementation, "Proc. of IEEE Int. Symposium on Intelligent Control, pp.277–282,1996.

[21] P. Gil, J. Henriques, A. Dourado and H. Duarte-Ramos, "Extended Neural Model Predictive Control of Non-linear Systems," Proceeding of IASTED –ASC 2000, pp. 94-100, 2000.

[22] M. Leskens , J. L. F. van Kessel, P. M. J. van den Hof and O. H Bosgra, "Nonlinear Model Predictive Control with Moving Horizon State and Disturbance Estimation – with Application to MSW Combustion," Proceedings of 16th IFAC World Congress, pp. 1-6, 2005.

[23] M. Nogaard, O. Ravn, N. Poulsen and L. K. Hansen, Neural Networks for Modelling and Control of Dynamic Systems, Springer-Verlag, 2003.

[24] M. Terziyska, "Newton optimization algorithm for nonlinear predictive controller," In Proc. of the Int. Conf. Automatics and Informatics'2005, Sofia, Bulgaria, pp. 247-250, 2005.

[25] W. Ray, Advanced Proceess Control, McGraw-Hill, New York, 1981