

NEURAL NETWORK PREDICTIVE CONTROL OF A CHEMICAL REACTOR

A. Vasičkaninová and M. Bakošová

Department of Information Engineering and Process Control,
Institute of Information Engineering, Automation and Mathematics,
Faculty of Chemical and Food Technology, Slovak University of Technology,
Radlinského 9, 81237 Bratislava, Slovakia
E-mail: {anna.vasickaninova, monika.bakosova}@stuba.sk

KEYWORDS

Model predictive control, neural network, continuous stirred tank reactor.

ABSTRACT

Model Predictive Control (MPC) refers to a class of algorithms that compute a sequence of manipulated variable adjustments in order to optimize the future behaviour of a plant. MPC technology can now be found in a wide variety of application areas. The neural network predictive controller that is discussed in this paper uses a neural network model of a nonlinear plant to predict future plant performance. The controller calculates the control input that will optimize plant performance over a specified future time horizon. In the paper, simulation of the neural network based predictive control for the continuous stirred tank reactor is presented. The simulation results are compared with fuzzy and PID control.

INTRODUCTION

Conventional process control systems utilize linear dynamic models. For highly nonlinear systems, control techniques directly based on nonlinear models can be expected to provide significantly improved performance.

Model Predictive Control (MPC) concept has been extensively studied and widely accepted in industrial applications. The main reasons for such popularity of the predictive control strategies are the intuitiveness and the explicit constraint handling.

The predictive controllers are used in many areas, where high-quality control is required (Qin and Badgwell, 1996; Qin and Badgwell, 2000; Rawlings, 2000). Model-based predictive control refers to a class of control algorithms, which are based on a process model. MBPC can be applied to such systems as e.g. multivariable, non-minimum-phase, open-loop unstable, non-linear, or systems with long time delays. Constrained model predictive control becomes the standard algorithm for advanced control in process industries.

Several versions of MPC techniques are Model Algorithmic Control (MAC) (Richalet et al., 1978),

Dynamic Matrix Control (DMC) (Cutler and Ramaker, 1980), and Internal Model Control (IMC) (Garcia and Morari, 1982). Although the above techniques differ from each other in some details, they are fundamentally the same, because all of them are based on linear process modelling. If the nonlinear model is available, the computational requirements are expected to be very high (Garcia et al., 1989), especially for nonlinear MIMO processes. It is estimated that, in a typical commissioning project, modelling efforts can take up to 90% of the cost and time in implementing a model predictive controller (Morari and Lee, 1999).

The Neural Network Model Predictive Control (NNMPC) is another typical and straightforward application of neural networks to nonlinear control. When a neural network is combined with MPC approach, it is used as a forward process model for the prediction of process output (Hunt et al., 1992; Nørgaard et al., 2000).

Control of chemical reactors is one of the most studied areas of process control. In this paper, a neural network based predictive control strategy is applied to a continuous-time stirred reactor with two parallel first-order irreversible exothermic reactions. Simulation results show that neural network based predictive control gives promising results.

MODEL-BASED PREDICTIVE CONTROL (MBPC)

MBPC is a name for several different control techniques. They all are associated with the same idea. The prediction is based on the model of the process.

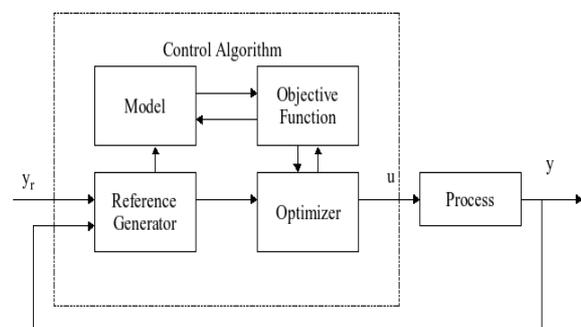


Figure 1: Classical model-based predictive control scheme.

The target of the model-based predictive control is to predict the future behaviour of the process over a certain horizon using the dynamic model and obtaining the control actions to minimize a certain criterion, generally

$$J(k, u(k)) = \sum_{j=N_1}^{N_2} (y(k+j) - y_r(k+j))^2 + \lambda \sum_{j=1}^{N_u} (u(k+j-1))^2 \quad (1)$$

Signals $y(k+j)$, $y_r(k+j)$, $u(k+j)$ are j -step ahead predictions of the process output, the reference trajectory and the control signal, respectively. The values N_1 and N_2 are minimal and maximal prediction horizons and N_u is the prediction horizon of control signal. The value of N_2 should cover the important part of the step response curve. The use of the control horizon N_u reduces the computational load of the method. The parameter λ represents the weight of the control signal. At each sampling period only the first control signal of the calculated sequence is applied to the controlled process. At the next sampling time the procedure is repeated. This is known as the receding horizon concept.

The controller consists of the plant model and the optimization block.

Eq. (1) is used in combination with input and output constraints:

$$\begin{aligned} u_{min} &\leq u \leq u_{max} \\ \Delta u_{min} &\leq \Delta u \leq \Delta u_{max} \\ y_{min} &\leq y \leq y_{max} \\ \Delta y_{min} &\leq \Delta y \leq \Delta y_{max} \end{aligned} \quad (2)$$

The ability to handle constraints is one of the key properties of MBPC and also caused its spread, use, and popularity in industry.

MBPC algorithms are reported to be very versatile and robust in process control applications.

NEURAL NETWORK PREDICTIVE CONTROL

Neural networks have been applied very successfully in the identification and control of dynamic systems. The universal approximation capabilities of the multilayer perceptron make it a popular choice for modelling of nonlinear systems and for implementing of nonlinear controllers.

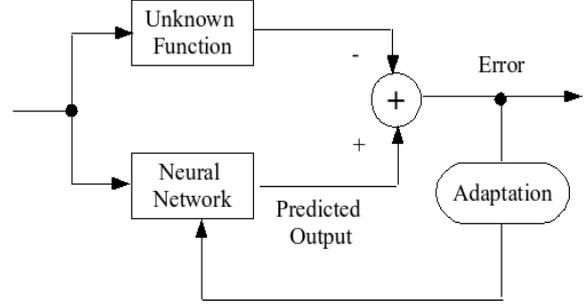


Figure 2: Neural network as a function approximator.

The unknown function may correspond to a system we are trying to control, in which case the neural network will be the identified plant model.

Two-layer networks, with sigmoid transfer functions in the hidden layer and linear transfer functions in the output layer, are universal approximators (Figure 2).

The prediction error between the plant output and the neural network output is used as the neural network training signal. The neural network plant model uses previous inputs and previous plant outputs to predict future values of the plant output. The structure of the neural network plant model is given in the following figure, where $u(k)$ is the system input, $y_p(k)$ is the plant output, $y_m(k)$ is the neural network model plant output, the blocks labelled TDL are tapped delay lines that store previous values of the input signal, $IW^{i,j}$ is the weight matrix from input number j to layer number i . $LW^{i,j}$ is the weight matrix from layer number j to layer number i .

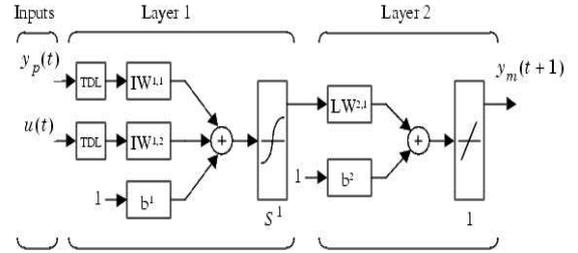


Figure 3: Structure of the neural network plant model.

This network can be trained off-line in batch mode, using data collected from the operation of the plant.

The procedure for selecting the network parameters is called training the network. The Levenberg-Marquardt (LM) algorithm is very efficient for training. The LM algorithm is an iterative technique that locates the minimum of a function that is expressed as the sum of squares of nonlinear functions. It has become a standard technique for nonlinear least-squares problems and can be thought of as a combination of steepest descent and the Gauss-Newton method (Levenberg, 1944; Madsen *et al.*, 2004; Marquardt, 1963; Mittelmann, 2004, Kelley, 1999).

When the current solution is far from the correct one, the algorithm behaves like a steepest descent method: slow, but guaranteed to converge. When the current solution is close to the correct solution, it becomes a Gauss-Newton method.

Let f be an assumed functional relation which maps a parameter vector $p \in R^m$ to an estimated measurement vector $\hat{x} = f(p)$, $\hat{x} \in R^n$. An initial parameter estimate p_0 and a measured vector x are provided and it is desired to find the vector p^+ that best satisfies the functional relation f , i. e. minimizes the squared distance $e^T e$ with $e = x - \hat{x}$. The basis of the LM algorithm is a linear approximation to f in the neighbourhood of p . For a small $\|\delta_p\|$, a Taylor series expansion leads to the approximation $f(p + \delta_p) \approx f(p) + J\delta_p$, where J is the Jacobian matrix $\frac{\partial f(p)}{\partial p}$. Like all non-linear optimization methods, LM is iterative: initiated at the starting point p_0 , the method produces a series of vectors p_1, p_2, \dots , that converge towards a local minimizer p^+ for f . Hence, at each step, it is required to find the δ_p that minimizes the quantity $\|e - J\delta_p\|$. The sought δ_p is thus the solution of a linear least-square problem: the minimum is attained when $J\delta_p - e$ is orthogonal to the column space of J . This leads to $J^T(J\delta_p - e) = 0$, which yields δ_p as the solution of the normal equations:

$$J^T J \delta_p = J^T e. \quad (3)$$

The matrix $J^T J$ in the left hand side of Eq. (3) is the approximate Hessian, i.e. an approximation to the matrix of second order derivatives. The LM method actually solves a slight variation of Eq. (3), known as the augmented normal equations $N\delta_p = J^T e$, where the off-diagonal elements of N are identical to the corresponding elements of $J^T J$ and the diagonal elements are given by $N_{ii} = \mu + [J^T J]_{ii}$ for some $\mu > 0$.

The strategy of altering the diagonal elements of $J^T J$ is damping and μ is referred to the damping term. If the updated parameter vector $p + \delta_p$ with δ_p computed from Eq. (3) leads to a reduction of the error e , the update is accepted and the process repeats with a decreased damping term. Otherwise, the damping term is increased, the augmented normal equations are solved again and the process iterates until a value δ_p of that decreases error is found.

In LM, the damping term is adjusted at each iteration to assure a reduction in the error e . The LM algorithm terminates when at least one of the following conditions is met:

- The magnitude of the gradient of $e^T e$, i.e. $J^T e$ in the right hand side of Eq. (3), drops below a threshold $\varepsilon \varepsilon_1$.

- The relative change in the magnitude of δ_p drops below a threshold ε_2 .
- The error $e^T e$ drops below a threshold ε_3 .
- A maximum number of iterations k_{max} is completed.

If a covariance matrix Σ for the measured vector x is available, the minimum is found by solving a weighted least squares problem defined by the weighted normal equations

$$J^T \Sigma J \delta_p = J^T \Sigma e \quad (4)$$

FUZZY CONTROL

Classic control theory is usually based on mathematical models which describe the behaviour of the process under consideration. The main aim of fuzzy control is to simulate a human expert (operator), who is able to control the process by translating the linguistic control rules into a fuzzy set theory.

In 1965 Lotfi A. Zadeh introduced fuzzy sets, where a more flexible sense of membership is possible. The past few years have witnessed a rapid growth in the use of fuzzy logic controllers for the control of processes that are complex and badly defined. Most fuzzy controllers developed till now have been of the rule-based type (Driankov et al., 1993), where the rules in the controller attempt to model the operator's response to particular process situations. An alternative approach uses fuzzy or inverse fuzzy model in process control (Babuška et al., 1995; Jang 1995) because it is often much easier to obtain information on how a process responds to particular inputs than to record how, and why, an operator responds to particular situations.

A review of the work on fuzzy control has been presented by Lee (Lee 1990).

Design of a simple fuzzy controller can be based on a three step design procedure, that builds on PID control: start with a PID controller, insert an equivalent, linear fuzzy controller and make it gradually nonlinear. A fuzzy controller can include empirical rules, and that is especially useful in operator controlled plants. Take e.g. a typical fuzzy controller:

- if error is negative and change in error is negative then output is negative big,
- if error is negative and change in error is zero then output is negative medium.

The collection of rules is called a rule base. The computer is able to execute the rules and compute a control signal depending on the measured inputs error and change in error. The inputs are most often hard or crisp measurements from some measuring equipment.

A dynamic controller would have additional inputs, for example derivatives, integrals, or previous values of measurements backwards in time.

The block fuzzification converts each piece of input data to degrees of membership by a lookup in one or several membership functions.

The rules may use several variables both in the condition and the conclusion of the rules. Basically a

linguistic controller contains rules in the if-then format, but they can be presented in different formats.

The resulting fuzzy set must be converted to a number that can be sent to the process as a control signal. This operation is called defuzzification. There are several defuzzification methods. Output scaling is also relevant. In case the output is defined on a standard universe this must be scaled to engineering units.

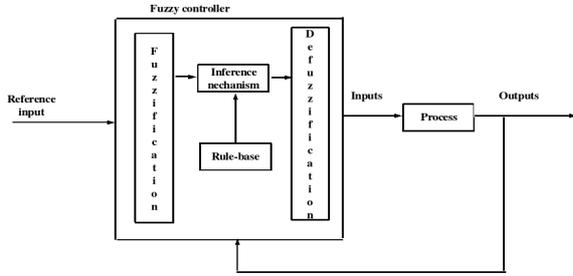


Figure 4: Fuzzy controller

Takagi-Sugeno type controller

The output sets can often be linear combinations of the inputs, or even a function of the inputs. The developed Fuzzy Logic Toolbox for the software package Matlab implements one of the hybrid schemes known as the Adaptive Network based Fuzzy Inference System (Anfis). Anfis represents a Sugeno-type fuzzy system in a special five-layer feed forward network architecture and uses a hybrid learning algorithm to identify the membership function parameters of single-output, Sugeno type fuzzy inference systems. Suppose the rule base of a Sugeno - Takagi fuzzy system is as follows (Nauck *et al.*, 1977; Takagi *et al.*, 1985.):

if x_1 is A_i and x_2 is B_i then $y = p_i x_1 + q_i x_2 + r_i$, $i = 1, \dots, N$ (5)

The if-parts (antecedents) of the rules describe fuzzy regions in the space of input variables error e , its derivative de and the then-parts (consequents) are functions of the inputs, usually linear with consequent parameters p_i , q_i , r_i , y is an output variable, A_i , B_i are fuzzy sets characterized by three linguistic variables (small, middle, large).

SIMULATIONS AND RESULTS

Chemical reactor

Consider a continuous stirred tank reactor (CSTR) (Vasičkaninová and Bakošová, 2005; Vasičkaninová and Bakošová, 2006) with two parallel first-order irreversible reactions according to the scheme

$A \xrightarrow{k_1} B \quad A \xrightarrow{k_2} C$. The measured and controlled

output is the temperature of the reaction mixture. The control input is the volumetric flow rate of the cooling medium.

The simplified non-linear dynamic mathematical model of the chemical reactor consists of five differential equations:

$$\frac{dc_A}{dt} = \frac{q}{V} c_{Av} - \frac{q}{V} c_A - k_1 c_A - k_2 c_A \quad (6)$$

$$\frac{dc_B}{dt} = \frac{q}{V} c_{Bv} - \frac{q}{V} c_B + k_1 c_A \quad (7)$$

$$\frac{dc_C}{dt} = \frac{q}{V} c_{Cv} - \frac{q}{V} c_C + k_2 c_A \quad (8)$$

$$\frac{dT}{dt} = \frac{q}{V} T_v - \frac{q}{V} T - \frac{Ak}{V\rho C_p} [T - T_c] + \frac{\dot{Q}_r}{V\rho C_p} \quad (9)$$

$$\frac{dT_c}{dt} = \frac{q_c}{V_c} T_{vc} - \frac{q_c}{V_c} T_c + \frac{Ak}{V_c \rho_c C_{pc}} [T - T_c] \quad (10)$$

The reaction rate coefficients are non-linear functions of the reaction temperature being defined by the Arrhenius relations

$$k_1 = k_{10} e^{-\frac{E_1}{RT}} \quad k_2 = k_{20} e^{-\frac{E_2}{RT}} \quad (11)$$

The heat generated by chemical reactions is expressed as

$$\dot{Q}_r = k_1 c_A V (-\Delta_r H_1) + k_2 c_A V (-\Delta_r H_2) \quad (12)$$

Here, c are concentrations, T are temperatures, V are volumes, ρ are densities, C_p are specific heat capacities, q are volumetric flow rates, $\Delta_r H$ are reaction enthalpies, A is the heat transfer area, k is the heat transfer coefficient. The subscript c denotes the coolant, r the reacting mixture and the superscript s denotes the steady-state values in the main operating point. Parameters and inputs of the reactor are enumerated in Table 1.

Table 1: Reactor parameters and inputs

$q = 0.015 \text{ m}^3 \text{ min}^{-1}$	$\Delta_r H_1 = -8.6 \cdot 10^4 \text{ kJ kmol}^{-1}$
$V = 0.23 \text{ m}^3$	$\Delta_r H_2 = -1.82 \cdot 10^4 \text{ kJ kmol}^{-1}$
$V_c = 0.21 \text{ m}^3$	$c_{Av} = 4.22 \text{ kmol m}^{-3}$
$\rho = 1020 \text{ kg m}^{-3}$	$c_{Bv} = 0 \text{ kmol m}^{-3}$
$\rho_c = 998 \text{ kg m}^{-3}$	$c_{Cv} = 0 \text{ kmol m}^{-3}$
$C_p = 4.02 \text{ kJ kg}^{-1} \text{ K}^{-1}$	$T_v = 328 \text{ K}$
$C_{pc} = 4.182 \text{ kJ kg}^{-1} \text{ K}^{-1}$	$T_{cv} = 298 \text{ K}$
$A = 1.51 \text{ m}^2$	$q^s_c = 0.004 \text{ m}^3 \text{ min}^{-1}$
$k = 42.8 \text{ kJ m}^{-2} \text{ min}^{-1} \text{ K}^{-1}$	$T^s = 363.61 \text{ K}$
$k_{10} = 1.55 \cdot 10^{11} \text{ min}^{-1}$	$T^s_c = 350.15 \text{ K}$
$k_{20} = 4.55 \cdot 10^{25} \text{ min}^{-1}$	$c^s_A = 0.4915 \text{ kmol m}^{-3}$

$E_1/R=9850 \text{ K}$	$c^s_B = 2.0042 \text{ kmol m}^{-3}$
$E_2/R=22019 \text{ K}$	$c^s_C = 1.7243 \text{ kmol m}^{-3}$

The reactions in the described reactor are exothermic ones and the heat generated by the chemical reactions is removed by the coolant in the jacket of the tank. The control objective is to keep the temperature of the reacting mixture close to a desired value.

Neural Network Model Predictive Control of the CSTR

The designed controller uses a neural network model to predict future CSTR responses to potential control signals. An optimization algorithm then computes the control signals that optimize future plant performance. The neural network plant model was trained using the Levenberg-Marquardt algorithm. The training data were obtained from the nonlinear model of the CSTR (6)-(10).

The used model predictive control method was based on the receding horizon technique. The neural network model predicted the plant response over a specified time horizon. The predictions were used by a numerical optimization program to determine the control signal that minimizes the following performance criterion over the specified horizon.

$$J(k, u(k)) = \sum_{j=N_1}^{N_2} (y_m(k+j) - y_r(k+j))^2 + \lambda \sum_{j=1}^{N_u} (u'(k+j-1) - u'(k+j-2)) \quad (13)$$

where N_1 , N_2 , and N_u define the horizons over which the tracking error and the control increments are evaluated. The u' variable is the tentative control signal, y_r is the reference response, and y_m is the network model response. The λ value determines the contribution that the sum of the squares of the control increments has on the performance index (Demuth and Beale, 2002). The controller block was implemented in Simulink. Constraints and parameters values: $0 \leq u \leq 0.02$, $354 \leq y_p \leq 365$, $N_1 = 1$, $N_2 = 7$, $N_u = 3$, $\lambda = 0.5$.

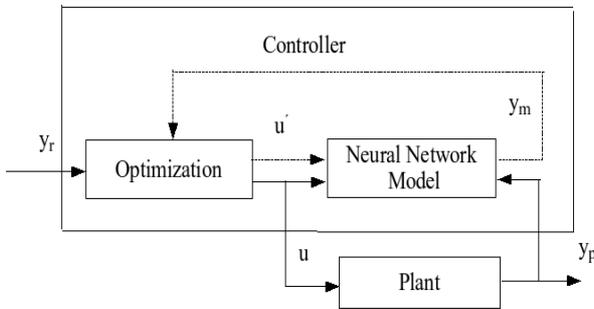


Figure 5: Neural Network Predictive Control.

Model based predictive control was simulated and the comparison with a fuzzy controller and PID controller was also made.

Takagi-Sugeno controller for the CSTR

Sugeno-type fuzzy inference system was generated using subtractive clustering in the form:

$$\text{if } e \text{ is } A_i \text{ and } de \text{ is } B_i \text{ then } u = p_i e + q_i de + r_i, \quad i=1, \dots, 3 \quad (14)$$

where e is the control error, de is the derivation of the control error, u is the calculated control input $q_c(t)$ and p_i , q_i , r_i are consequent parameters. The symmetric Gaussian function (*gaussmf* in MATLAB) was chosen as the membership function and it depends on two parameters σ and c as it is seen in (14)

$$f(x; \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}} \quad (15)$$

The parameters σ and c for *gaussmf* are listed in the Table 2. For obtaining of these parameters, it was necessary to have the data sets of e , de and u at first. These data were obtained by simulation of PID control of the CSTR. The consequent parameters in the control input rule (14) are listed in Table 3 and the resulting plot of the output surface of a described fuzzy inference system is presented in Figure 6.

Table 2: Gaussian curve membership functions parameters.

e		de	
σ_i	c_i	σ_i	c_i
0,52	0,0018	0,53	0,0345
0,52	-0,0839	0,53	0,0003
0,52	-0,0665	0,53	0,0066

Table 3: Consequent parameters

p_i	q_i	r_i
-0,256	-0,092	4,04
-3,910	-1,299	-156,12
-6,562	-2,517	151,33

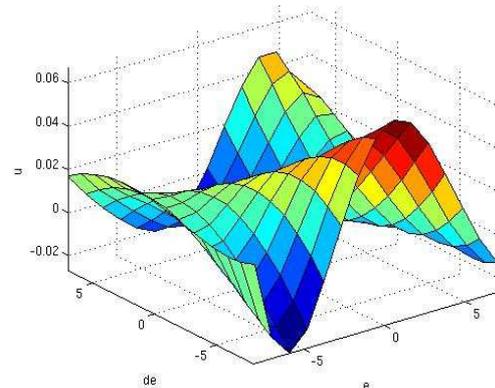


Figure 6: Takagi-Sugeno controller - control signal u as function of control error e and its derivation de .

The feedback PID controllers were tuned by various methods (Ogunnaike and Ray, 1994) and the best simulation results were obtained with PID controller tuned using Chien-Hrones-Reswick method and its parameters are $K_C = -0.003$, $T_I = 16.8$, $T_D = 1.41$.

Figure 7 presents the simulation results of the predictive control of the CSTR. These results are compared with those obtained by fuzzy control and PID control of the CSTR.

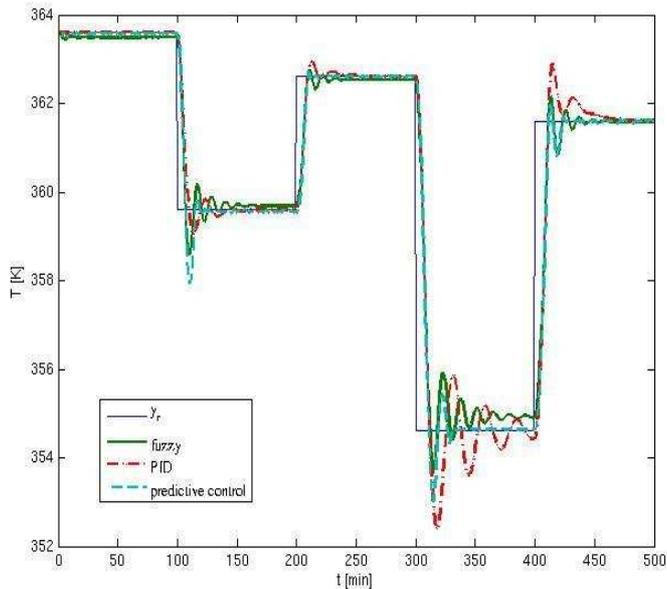


Figure 7: Comparison of fuzzy and neural network predictive control of a CSTR with PID control.

The step changes of the reference y_r were generated and the MBP, fuzzy and PID controllers were compared using the well-known iae criterion described as follows:

$$iae = \int_0^T (e) dt \quad (16)$$

The iae values are given in Table 4.

Table 4: Comparison of the simulation results by integrated absolute error.

control method	iae
predictive control	174
fuzzy control	208
PID control	219

CONCLUSIONS

In this paper, an application of a neural network based predictive control strategy to a CSTR is presented. The simulation results confirmed that neural network based predictive control is one of the possibilities for successful control of CSTRs. The advantage of this

approach is that it is not linear-model-based strategy and the control input constraints are directly included to the synthesis. Comparison of the MBPC simulation results with fuzzy control and classical PID control demonstrates the effectiveness and superiority of the proposed approach. Used fuzzy controller is simple, but it can lead to the nonzero steady-state control errors.

REFERENCES

- Babuška R, Sousa J, Verbruggen, H, B, 1995. Model-Based Design of Fuzzy Control Systems. *Proceedings of the EUFIT'95*, 837-841
- Cutler C. R. and Ramaker, B. L. 1980. Dynamic matrix control - a computer control algorithm. *Proc. Amer. Control Conf.*, WP5-B
- Demuth, H. and Beale, M. 2002. Neural Network Toolbox For Use with MATLAB®. Control Systems. 205-242. URL http://www.image.ece.ntua.gr/courses_static/nn/matlab/nnet.pdf.
- Driankov D, Hellendorn, H. 1993. *An Introduction to Fuzzy Control*. Springer Verlag, London
- Garcia C. E. and Morari, M. 1982. Internal model control-I. A unifying review and some new results. *Ind. Eng. Chem. Process Des. Dev.* **21**, 308--323.
- Garcia C. E., D. M. Prett and Morari, M. 1989. Model predictive control, theory and practice-a survey. *Automatica* **25**(3), 335-348.
- Hunt K. J., D. Sbarbaro, R. Zbikowski and Gawthrop, P. J. 1992. Neural networks for control systems-A survey. *Automatica* **28**(6), 1083-1112.
- Jang J. S. R., Sun, C. T. 1995. Neuro-Fuzzy Modeling and Control. *The Proceedings of the IEEE May 83*, 378-406.
- Kelley, C. T. 1999. *Iterative Methods for Optimization*. SIAM Press, Philadelphia.
- Lee, C. C. 1990. Fuzzy Logic in Control Systems: Fuzzy Logic Controllers, *IEEE Trans. Syst. Mn. Cybern.*, **SMC 20**, p. 404.
- Levenberg, K. 1944. A Method for the Solution of Certain Non-linear Problems in Least Squares, *Quarterly of Applied Mathematics*, **2**(2), 164-168.
- Madsen, K., H. B. Nielsen and Tingleff, O. 2004. *Methods for Non-Linear Least Squares Problems*. Technical University of Denmark, Lecture notes, Available: <http://www.imm.dtu.dk/courses/02611/nllsq.pdf>.
- Marquardt, W. 1963. An Algorithm for the Least-Squares Estimation of Nonlinear Parameters. *SIAM Journal of Applied Mathematics*, **11**(2), 431-441.
- Mittelmann, D. 2004. The Least Squares Problem. Available: <http://plato.asu.edu/topics/problems/nllsq.html>.
- Morari, M. and Lee, J. H. 1999. Model predictive control: past, present and future, *Computers and Chemical Engineering*, **23**, 667-682.
- Nauck, D., Klawonn, F. and Kruse, R. 1977. *Foundations of neuro-fuzzy systems*. John Wiley & Sons, Great Britain. Able, B.C. (1956). Nucleic acid content of microscope. *Nature* **135**, 7-9

- Nørgaard, M., Ravn, O., Poulsen, N. K. and Hansen, L. K. 2000. *Neural networks for modelling and control of dynamic systems*, Springer, London
- Ogunnaike, B. A. and Ray, W. H. 1994. *Process Dynamics, Modelling, and Control*. 536-541. Oxford University Press. New York. 1259. ISBN 0-19-509119-1.
- Qin, S. J. and Badgwell, T. A. 1996. An overview of industrial model predictive control technology. *Fifth International Conference on Chemical Process Control - CPC V*, 232-56. American Institute of Chemical Engineers.
- Qin, S. J. and Badgwell, T. A. 2000. An overview of nonlinear model predictive control applications. *Nonlinear Predictive Control*, 369-393. Birkhäuser.
- Rawlings, J. B. 2000. Tutorial overview of model predictive control. *IEEE Contr. Syst. Magazine*, **20**(3), 38-52.
- Richalet J. A., Rault, A., Testud, J. D. and Papon, J. 1978. Model predictive heuristic control: applications to an industrial process. *Automatica* **14**, 413-428.
- Takagi K, Sugeno M. 1985. Fuzzy identification of systems and its applications to modelling and control. *IEEE Trans. Syst. Man. Cybern.* **15**, 116-132.
- Vasičkaninová, A. and Bakošová, M. 2005. Cascade fuzzy logic control of a chemical reactor. In *Proc. 15. Int. Conference Process Control '05*, Štrbské Pleso, High Tatras.
- Vasičkaninová, A. and Bakošová, M. 2006. Fuzzy modelling and identification of the chemical technological processes. In *Proc. 7. Int. Scientific-*

Tehnickal Conf. Process Control 2006, University of Pardubice, Kouty nad Desnou.

- Zadeh, L. A. 1965. Fuzzy Sets. *Information and Control*, Vol.8. pp. 338-353. Neural Network Toolbox™ - Getting Started Guide, The MathWorks, 2008

ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/4055/07 and 1/0071/09.

AUTHOR BIOGRAPHIES

ANNA VASIČKANINOVÁ studied at the Slovak University of Technology in Bratislava, where she got her master degree in chemical and process engineering. Her research interest are modeling and simulation of chemical processes, system identification and predictive, fuzzy and neuro-fuzzy control. You can contact she on email address anna.vasickaninova@stuba.sk.

MONIKA BAKOŠOVÁ graduated at the Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava in 1982. She received her PhD. degree in 1987 on hierarchical control of large-scale systems. In 1999 he was appointed Associate Professor at the FCFT. Since 1986 she is with the STU in Bratislava. Her research interests are focused on modeling, simulation and control of processes in chemical technology with special emphasis for the problems of robust stabilization and robust control. Her e-mail address is: monika.bakosova@stuba.sk.