Acta Chimica Slovaca, Vol.2, No.2, 2009, 21 - 36

Neural Network Predictive Control of a Chemical Reactor

Anna Vasičkaninová*, Monika Bakošová

Institute of Information Engineering, Automation and Mathematics,
Faculty of Chemical and Food Technology, Slovak University of Technology,
Radlinského 9, 81237 Bratislava, Slovakia

monika.bakosova@stuba.sk

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 of the continuous stirred tank reactor is presented. The simulation results are compared with fuzzy and PID control.

Keywords: model predictive control, fuzzy control, PID control, neural network, continuous stirred tank reactor

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, see e.g. 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), see e.g. Richalet et al. (1978), Dynamic Matrix Control (DMC), see e.g. Cutler et al. (1980), and Internal Model Control (IMC), see e.g. Garcia et al. (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 very high, see e.g. 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 by 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, see e.g. 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 the neural network based predictive control gives promising results.

Theoretical

Model-based predictive control

MBPC is a name of several different control techniques. All are associated with the same idea. The prediction is based on the model of the process, as it is shown in Figure 1.

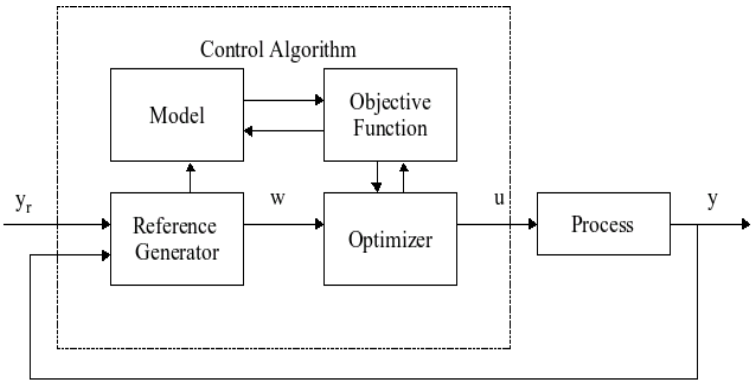

Fig. 1. Classical model-based predictive control scheme

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

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

Signals $y_m(k+j)$, $y_r(k+j)$, $u(k+j)$ are the j-step ahead predictions of the process output, the reference trajectory and the control input, respectively. The values N_1 and N_2 are the minimal and maximal prediction horizon of the controlled output, and N_u is the prediction horizon of the control input. 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 the input and output constraints:

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

The ability to handle constraints is one of the key properties of MBPC and also causes 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. The use of a neural network for process modelling is shown in Figure 2. The unknown function may correspond to a controlled system, and the neural network is 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.

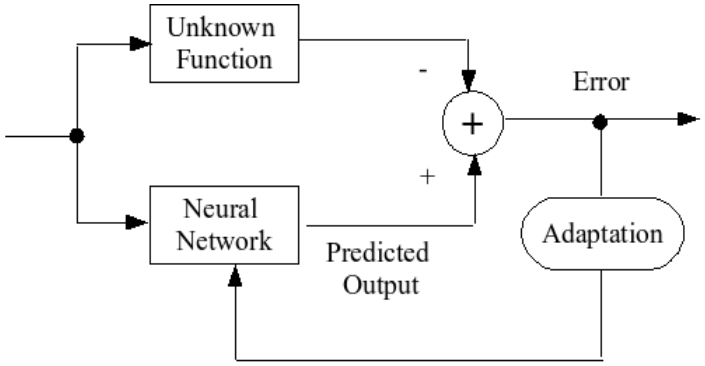

Fig. 2. Neural network as a function approximator

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 Figure 3, where $u(t)$ is the system input, $y_p(t)$ is the plant output, $y_m(t)$ is the neural network model plant output, the blocks labelled TDL are tapped delay lines that store previous values of the input signal, $\boldsymbol{IW}^{i,j}$ is the weight matrix from the input j to the layer i. $\boldsymbol{LW}^{i,j}$ is the weight matrix from the layer j to the layer i.

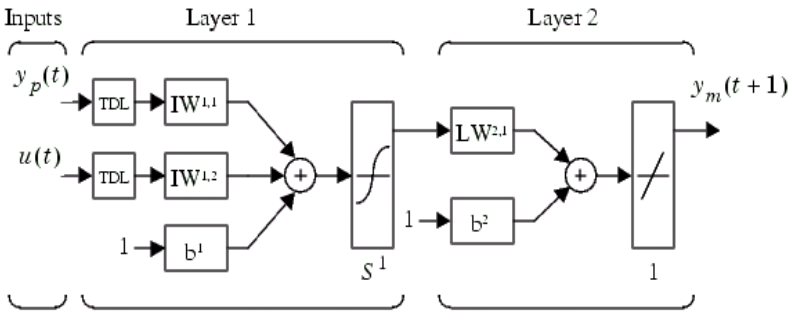

Fig. 3. Structure of the neural network plant model

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, see e. g. Kelley (1999), Levenberg (1944), Madsen et al. (2004), Marquardt (1963), Mittelmann (2004).

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 $\boldsymbol{p}_0$ and a measured vector $\boldsymbol{x}$ are provided, and it is desired to find the vector $\boldsymbol{p}^+$ 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 $\boldsymbol{p}$. For a small $\|\delta_p\|$, a Taylor series expansion leads to the approximation $f(p + \delta_p) \approx f(p) + J\delta_p$ where $\boldsymbol{J}$ is the Jacobi matrix $\frac{\partial f(p)}{\partial p}$. Like all non-linear optimization methods, LM is iterative: initiated at the starting point $\boldsymbol{p}_0$, the method produces a series of vectors $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, ..., that converge towards a local minimizer $\boldsymbol{p}^+$ 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 $\boldsymbol{J}$. This leads to $J^T(J\delta_p - e) = 0$, which yields δ_p as the solution of the normal equations:

$$J^T J\delta_p = J^T e. \tag{3}$$

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 of δ_p 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:

1. 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 ε_1.
2. The relative change in the magnitude of δ_p drops below a threshold ε_2.
3. The error $e^T e$ drops below a threshold ε_3.
4. A maximum number of iterations k_{max} is completed.

If a covariance matrix $\sum$ for the measured vector $\boldsymbol{x}$ is available, the minimum is found by solving a weighted least squares problem defined by the weighted normal equations

$$J^T \sum J\delta_p = J^T \sum e \tag{4}$$

Fuzzy Control

Classic control theory is usually based on mathematical models which describe the behaviour of the controlled process. 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 by 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, see e.g. 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 (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; make the controller gradually nonlinear. The fuzzy controller can include empirical rules. This property is especially useful in operator controlled plants. Let us consider 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.

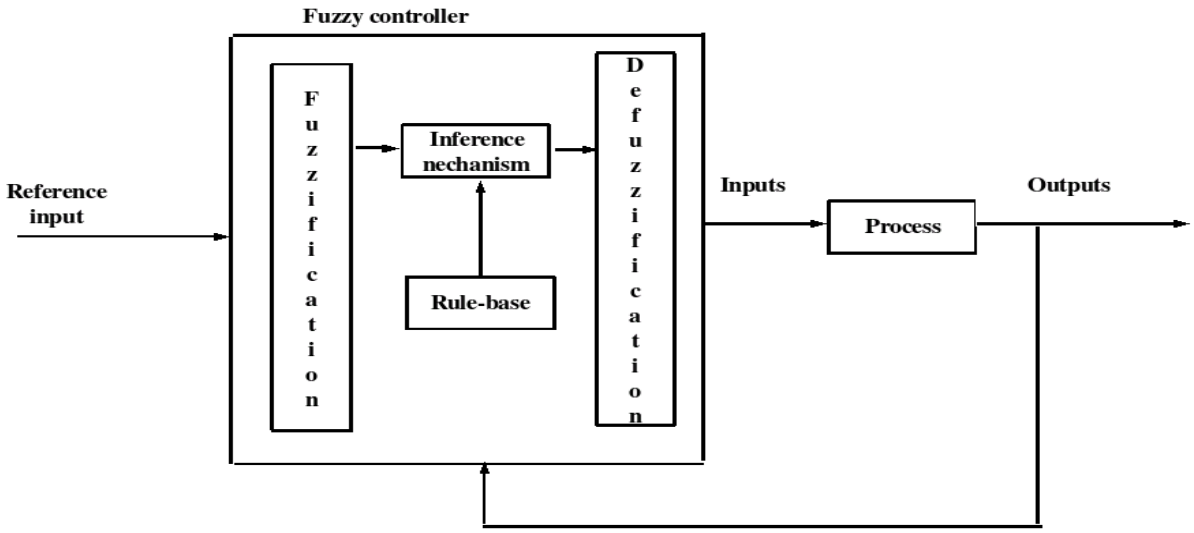

Fig. 4. Fuzzy controller

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 the 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, see e.g. Nauck et al. *(*1977), Takagi et al. (1985), Kvasnica et al. (2009):

$$\text{if } x_1 \text{ is } A_i \text{ and } x_2 \text{ is } B_i \text{ then } y = p_i x_1 + q_i x_2 + r_i,\ i=1,..N \tag{5}$$

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

Experimental

Consider a continuous stirred tank reactor (CSTR) by Vasičkaninová et al. (2005), Vasičkaninová et al. (2006) with two parallel first-order irreversible reactions according to the scheme $A \xrightarrow{k_1} B, A \xrightarrow{k_2} C$, where B is the main product and C is the side product. The measured and controlled output is the temperature of the reaction mixture. The control input is the volumetric flow rate of the cooling medium. Possible disturbances include changes in the feed temperature and the coolant temperature. The only manipulated variable is the coolant flow rate.

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

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

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

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

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

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

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

$$k_1 = k_{10}e^{-\frac{E_1}{RT}} \qquad k_2 = k_{20}e^{-\frac{E_2}{RT}} \tag{11}$$

The heat generated by chemical reactions is expressed as

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

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

Variable	Unit	Value
q	$m^3 min^{-1}$	0.015
V	m^3	0.23
V_C	m^3	0.21
ρ	kg m^{-3}	1020
ρ_C	kg m^{-3}	998
C_p	kJ kg^{-1} K^{-1}	4.02
C_{pc}	kJ kg^{-1} K^{-1}	4.182
A	m^2	1.51
k	kJ m^{-2} min^{-1} K^{-1}	42.8
k_{10}	min^{-1}	$1.55.10^{11}$
k_{20}	min^{-1}	$4.55.10^{25}$
E_1/R	K	9850
E_2/R	K	22019

Variable	Unit	Value
$\Delta_r H_1$	kJ $kmol^{-1}$	$-8.6.10^4$
$\Delta_r H_2$	kJ $kmol^{-1}$	$-1.82.10^4$
c_{Av}	kmol m^{-3}	4.22
c_{Bv}	kmol m^{-3}	0
c_{Cv}	kmol m^{-3}	0
T^s_v	K	328
T^s_{vc}	K	298
q^s_c	m^3 min^{-1}	0.004
T^s	K	363.61
T^s_c	K	350.15
c^s_A	kmol m^{-3}	0.4915
c^s_B	kmol m^{-3}	2.0042
c^s_C	kmol m^{-3}	1.7243

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.

Results and discussion

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 performance criterion (1) over the specified horizon. The controller block was implemented in Simulink. Constraints and parameters values: $0 \leq u \leq 0.02$, $354 \leq y_p \leq 365$, $N_1 = 1$, $N_2 = 7$, $N_u = 3$, $\lambda = 0.5$.

Takagi-Sugeno controller for the CSTR

Sugeno-type fuzzy inference system was generated using subtractive clustering in the form: if e is A_i and de is B_i then $u = p_i\, e + q_i\, de + r_i$, $i=1, \ldots 3$ (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)

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

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 5.

Table 2: Parameters of the Gaussian curve membership functions

e		*de*	
σ_i	c_i	σ_i	c_i
0.348	0.088	0.348	-0.025
0.348	-0.072	0.348	0.004
0.348	-0.048	0.348	0.003
0.348	0.281	0.348	-0.055

Table 3: Consequent parameters

p_i	q_i	r_i
-0.030	-0.127	-0.042
0.057	0.013	-0.848
0.124	-0.073	-0.780
-0.002	0.0	0.012

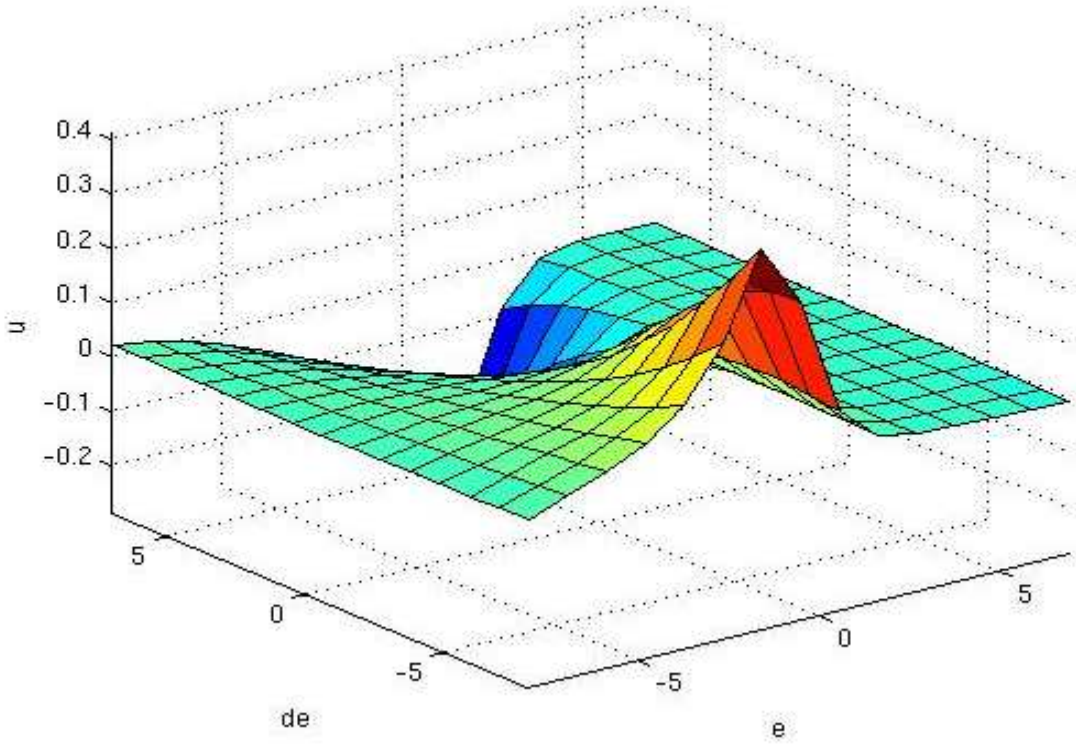

Fig. 5. Takagi-Sugeno controller - control signal *u* as function of control error *e* and its derivation *de*.

PID control

For feedback controller tuning, the approximate model of a system with complex dynamics can have the form of a first-order-plus-time-delay transfer function *(16*). The process is characterised by a steady-state gain *K*, an effective time constant *T* and an effective time delay *D*.

$$G_P(s) = \frac{K}{Ts+1} e^{-Ds} \qquad (16)$$

The transfer function describing the controlled chemical reactor was identified from step response data in the form (16) with parameters: K = -1257, T = 14 min, D = 2 min. These parameters were used for feedback controller tuning. The feedback PID controllers were tuned by various methods, see e.g. Ogunnaike and Ray (1994). The best simulation results were obtained with PID controller (17) tuned using Chien-Hrones-Reswick method. The controller parameters are K_C = *-0.003*, T_I = *16.8*, T_D = *1.41*. The transfer function of the used PID controller is following

$$G_C(s) = K_C\left(1 + \frac{1}{T_I s} + T_D s\right) \qquad (17)$$

Figure 6 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

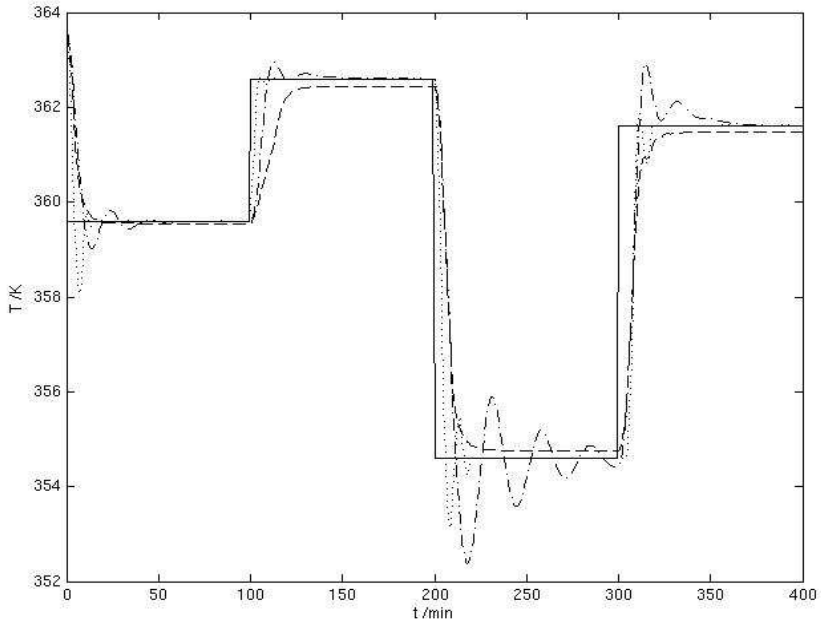

Fig. 6. Comparison of the reacting mixture temperature control: predictive control (.....), fuzzy control (- - -), PID control (- . - . -), reference trajectory (__)

The step changes of the reference y_r were generated and the MBP, fuzzy and PID controllers were compared using *iae* and *ise* criteria described as follows:

$$iae = \int_0^T e \, dt \tag{18}$$

$$ise = \int_0^T e^2 \, dt \tag{19}$$

The *iae* and *ise* values are given in Table 4.

Table 4: Comparison of the simulation results by integrated absolute error *iae* and integrated square error *ise*

control method	**iae**	**ise**
predictive control	168	728
fuzzy control	211	724
PID control	220	771

Figure 7 presents the simulation results of the predictive control, fuzzy control and PID control of the CSTR in the case when disturbances affect the controlled process. Disturbances were represented by coolant temperature changes from 298 K to 327 K at t=100 min, from 327 K to 291 K at t=300 min and from 291 K to 310 K at t=500 min. The *iae* and *ise* values are given in Table 5.

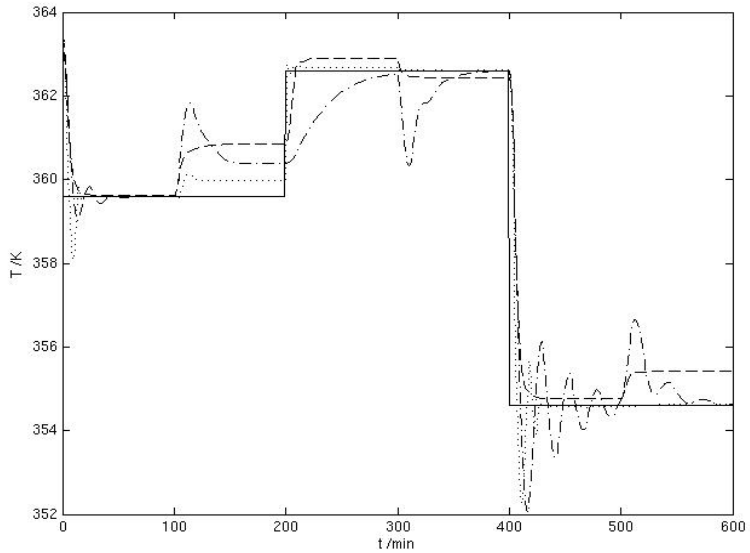

Fig. 7. Comparison of the reacting mixture temperature control in case when disturbances affect the controlled process: predictive control (.....), fuzzy control (- - -), PID control (-.-.-), reference trajectory (__)

Table 5: Comparison of the simulation results by integrated absolute error and integrated square error in case when disturbances affect the controlled process.

control method	***iae***	***ise***
predictive control	161	421
fuzzy control	322	712
PID control	405	772

Used fuzzy controller is simple, and it offers lesser value ise than the predictive controller in the case when the reactor is not affected by disturbances. The disadvantage of the fuzzy controller is, that using the controller leads to nonzero steady-state errors. The steady-state errors vary from 0.05 K to 0.17 K, when the reactor without disturbances is controlled. In the case of the reactor control in the presence of disturbances, the steady-state errors vary from 0.17 K to 1.24 K. The advantage of the fuzzy control is that the control responses do not show any overshoots and undershoots. The worst simulation results were obtained using the PID controller. The control responses are most oscillating, and the PID controller used in a simple feedback control loop is not able to attenuate disturbances. The best simulation results were obtained using the neural network predictive controller. Although the control responses are oscillating, the maximum overshoot is smaller than the one with the fuzzy controller. Simultaneously, the steady state errors are very small, the maximum steady state error is 0.38 K in the presence of disturbances. The followed integral criteria also confirm that the best of three controllers in the neural network predictive controller.

Conclusions

In this paper, an application of a neural network based predictive control strategy to a CSTR is presented. The simulation results confirm that the 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. These properties are apparent, especially in the case, when the controlled process is affected by disturbances.

Acknowledgments

The work has been supported by the Scientific Grant Agency of the Slovak Republic under grants 1/4055/07, 1/0071/09 and by the Slovak Research and Development Agency under the project APVV-0029-07.

References

Babuška R, Sousa J, Verbruggen HB (1995) Model-Based Design of Fuzzy Control Systems. *Proceedings of the EUFIT´95:* 837-841.

Cutler CR, Ramaker BL (1980) Dynamic matrix control - a computer control algorithm. *Proc. Amer. Control Conf.*, WP5-B.

Demuth H. 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, Hellendom H (1993) *An Introduction to Fuzzy Control*. Springer Verlag, London.

Garcia CE, Morari M (1982) Internal model control-I. A unifying review and some new results. *Ind. Eng. Chem. Process Des. Dev*. **21**: 308--323.

Garcia CE, Prett DM, Morari M (1989) Model predictive control, theory and practice-a survey. *Automatica* **25**(3): 335-348.

Hunt KJ, Sbarbaro D, Zbikowski R, Gawthrop PJ (1992) Neural networks for control systems-A survey. *Automatica* **28**(6): 1083-I 112.

Jang JSR, Sun CT (1995) Neuro-Fuzzy Modeling and Control. *The Proceedings of the IEEE May 83:* 378-406.

Kelley CT (1999) Iterative Methods for Optimization. SIAM Press, Philadelphia.

Kvasnica M, Herceg M, Čirka Ľ, Fikar M (2009) Time-Optimal Control of Takagi-Sugeno Fuzzy Systems. *Proceedings of the 10th European Control Conference:* 916–921.

Lee CC (1990) Fuzzy Logic in Control Systems: Fuzzy Logic Controllers, I*EEE Trans. Syst. Mn. Cybern., SMC* **20**: 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, Nielsen HB, 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/ nlolsq. html.

Morari M, Lee JH (1999) Model predictive control: past, present and future, *Computers and Chemical Engineering*, **23**: 667-682.

Nauck D, Klawonn F, 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 NK, Hansen L K (2000) *Neural networks for modelling and control of dynamic systems*, Springer, London.

Ogunnaike BA, Ray WH (1994) *Process Dynamics, Modelling, and Control*. 536-541. Oxford University Press. New York. 1259. ISBN 0-19-509119-1.

Qin SJ, Badgwell TA (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 SJ, Badgwell TA (2000) An overview of nonlinear model predictive control applications. *Nonlinear Predictive Control*: 369-393. Birkhäuser.

Rawlings JB (2000) Tutorial overview of model predictive control. *IEEE Contr. Syst. Magazine*, **20**(3): 38-52.

Richalet JA, Rault A, Testud JD, 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, 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, Bakošová M (2006) Fuzzy modelling and identification of the chemical technological processes. In *Proc. 7. Int. Scientific-Tehnical Conf. Process Control 2006*, University of Pardubice, Kouty nad Desnou.

Zadeh LA (1965) Fuzzy Sets. *Information and Control*, Vol.8: 338-353.

Neural Network Toolbox™ - Getting Started Guide, The MathWorks, 2008