

Artificial Neural Network – Possible Approach to Nonlinear System Control

Jan Mareš, Petr Doležel and Pavel Hrnčířík
*Institute of Chemical Technology, Prague
& University of Pardubice, Pardubice
Czech Republic*

1. Introduction

Artificial Neural Networks (ANN) have traditionally enjoyed considerable attention in process control applications. Thus, the paper is focused on real system control design using neural networks. The point is to show whether neural networks bring better performances to nonlinear process control or not.

Artificial Neural Network is nowadays a popular methodology with lots of practical and industrial applications. As introduction, some concrete examples of successful application of ANN can be mentioned, *e.g.* mathematical modeling of bioprocesses [Montague et al., 1994], [Teixeira et al., 2005], prediction models and control of boilers, furnaces and turbines [Lichota et al., 2010] or industrial ANN control of calcinations processes, or iron ore process [Dwarapudi, et al., 2007].

Specifically in our paper, the aim is to explain and describe usage of neural network in the case of nonlinear reactor furnace control.

2. Controlled system

Real system (controlled plant) is a reactor furnace, which is significantly nonlinear system. Furnace is an equipment of the research laboratory of the Department of Physical Chemistry at the University of Pardubice, Czech Republic.

Reactor furnace is used for research of oxidation and reduction qualities of catalyzers under different temperatures by controlled heating of the reactor (where the chemical substance is placed). The temperature profile of the reactor is strictly defined. It is linear increasing up to 800 °C, then keeping the constant value of 800 °C till the end of the experiment. The difference between the setpoint and controlled variable (furnace temperature) has to be less than 10 °C.

The basic premise is so strict, that it is not possible to use standard control techniques as PID controller. Thus, an artificial neural network represents one of the available techniques for overcoming this obstacle.

2.1 System description

Reactor furnace base is a cored cylinder made of insulative material, described in [Mareš et al., 2010a]. On the inner surface there are two heating spirals (powered by voltage 230 V). In

the middle of the cylinder there is a reactor. The reactor temperature is measured by one platinum thermometer (see Figure 1).

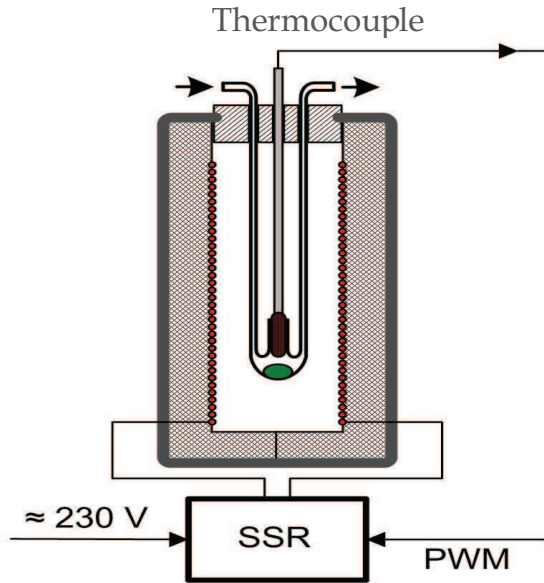


Fig. 1. Reactor furnace chart

The system is a thermal process with two inputs (spiral power and ambient temperature) and one output (reactor temperature). Thus, the controlled variable is the reactor temperature and the manipulated variable is the spiral power with the ambient temperature as measured error. The plant is significantly nonlinear system. Nonlinearity is caused by heat transfer mechanism. When the temperature is low, heat transfer is provided only by conduction. However, when the temperature is high, radiation presents an important transfer principle.

2.2 Nonlinear model

Nonlinear mathematical model of reactor furnace consists of four parts. Differential equations describing isolation, heating spiral, inner space and reactor were derived. Because variables changes along devices dimension are irrelevant, process behavior can be considered as a lumped system.

Nonlinear mathematical model is possible to describe by equations (1) to (4), more in [Mares et al., 2010a].

Isolation

$$\begin{aligned}
 & \underbrace{\alpha_{AB} \cdot S_{AB}}_{aAB} \cdot (T_B - T_A) + \underbrace{\alpha_{AC} \cdot S_{AC}}_{aAC} \cdot (T_C - T_A) + S_2 \sigma (T_B^4 - T_A^4) = \\
 & = \underbrace{\alpha_{AO} \cdot S_{AO}}_{aAO} \cdot (T_A - T_{OK}) + S_3 \sigma (T_A^4 - T_{OK}^4) + S_4 \sigma (T_A^4 - T_D^4) + \underbrace{m_A \cdot c_A}_{PA} \cdot \frac{dT_A}{dt}
 \end{aligned}
 \tag{1}$$

Spiral

$$\frac{E}{1 + \beta T_B} = \underbrace{\alpha_{AB} \cdot S_{AB}}_{aAB} \cdot (T_B - T_A) + \underbrace{\alpha_{BC} \cdot S_{BC}}_{aBC} \cdot (T_B - T_C) + S_1 \cdot \sigma \cdot (T_B^4 - T_D^4) + S_2 \cdot \sigma \cdot (T_B^4 - T_A^4) + \underbrace{m_B \cdot c_B}_{PB} \cdot \frac{dT_B}{dt} \quad (2)$$

Inner space

$$\underbrace{\alpha_{BC} \cdot S_{BC}}_{aBC} \cdot (T_B - T_C) = \underbrace{\alpha_{CO} \cdot S_{CO}}_{aCO} \cdot (T_C - T_{OK}) + \underbrace{\alpha_{CD} \cdot S_{CD}}_{aCD} \cdot (T_C - T_D) + \underbrace{\alpha_{AC} \cdot S_{AC}}_{aAC} \cdot (T_C - T_A) + \underbrace{m_C \cdot c_C}_{PC} \cdot \frac{dT_C}{dt} \quad (3)$$

Reactor

$$\underbrace{\alpha_{CD} \cdot S_{CD}}_{aCD} \cdot (T_C - T_D) + S_1 \cdot \sigma \cdot (T_B^4 - T_D^4) + S_4 \cdot \sigma \cdot (T_A^4 - T_D^4) = \underbrace{m_D \cdot c_D}_{PD} \cdot \frac{dT_D}{dt} \quad (4)$$

where

A is isolation

B is spiral

C is inner space

D is reactor

α_{ij} , $J \cdot K^{-1} \cdot m^{-2} \cdot s^{-1}$, is transfer coefficient between i and j

S_{ij} , m^2 , is surface of contact between i and j

S_1, S_2, S_3, S_4 are surfaces of reactor, isolation inside and outside surface of the furnace

m_i , kg , is weight of i

β , K^{-1} , is spiral temperature coefficient

c_i , $J \cdot K^{-1} \cdot kg^{-1}$, is capacity of i

σ , $J \cdot K^{-4} \cdot m^{-2} \cdot s^{-1}$, is Stefan-Bolzmann constant

From the model it is evident that the system is strongly nonlinear and very difficult to control. Thus complex techniques are necessary to use.

3. Control techniques

Several control techniques with neural network were chosen, applied and compared to classical ones. One of the objectives is to find out whether control techniques with neural networks bring any improvement to control performances at all. Brief description of the applied techniques is given below.

3.1 Internal model control

Standard internal model control (IMC) is technique closely connected to direct inverse control which brings some limitations to system to be controlled. On the other side, IMC has some convenient features, e.g. it is able to cope well with output disturbances. The concept of IMC is

presented in [Rivera et al., 1986]. IMC for nonlinear systems is introduced in [Economou et al., 1986] and IMC with neural networks is described e.g. in [Norgaard et al., 2000].

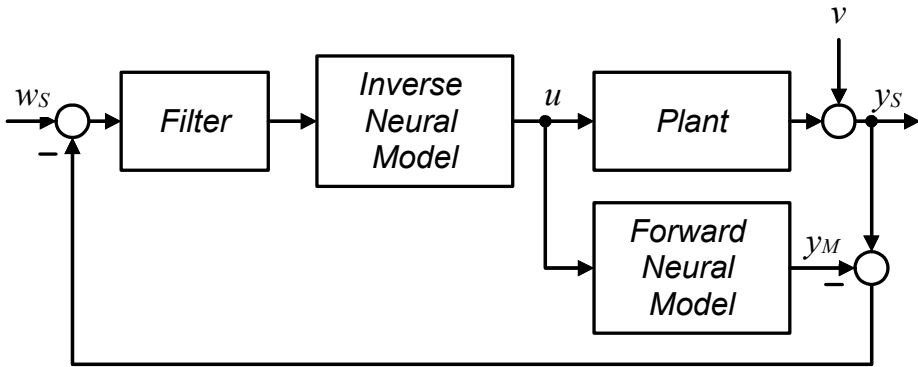


Fig. 2. Internal model control scheme

Internal model controller requires a forward model as well as an inverse model of the system to be controlled. Both models are replaced with adequate neural network model - design of both models is described in [Nguyen et al., 2003]. Then, control loop can be put together - see Fig. 2, where w_s , u , v , y_s , and y_M are reference variable, control signal, output disturbance, control variable and forward model output. It can be shown, that equation (5) is valid in case of ideal inverse and forward neural model. In some cases, filtering can be applied ahead of inverse controller to smooth reference variable to eliminate negative influence of sudden changes. In the case of linear continuous-time IMC, filter usage is essential.

$$y_s = 1w_s + 0v \tag{5}$$

The equation above is unattainable in real processes but can be approximately approached if discrete neural models are used.

In section 4.3, control experiments with neural models of linear IMC as well as IMC with neural models are demonstrated

3.2 Predictive control

Predictive control is used in two variants. The first one is typical Model Predictive Control and the second one is Neural Network Predictive Control.

3.2.1 Model predictive control

Model predictive control (MPC) is widely used technique for process control in industry, where better control performance is necessary. MPC is a general strategy which comes from the process model, therefore MPC controllers are truly-tailor-made. The working principle is briefly described in this chapter (the description is not in general, but only for SISO systems), more in [Camacho, 2007].

The mathematical model of the controlled system is assumed in the form of equation (6).

$$A(z^{-1})y(k) = z^{-d}.B(z^{-1})u(k-1) + C(z^{-1})e(k) \tag{6}$$

where A, B, C are polynomials, $y(k)$ is model output, $u(k)$ is model input $e(k)$ is output error. The model without errors and without output delay is supposed, therefore $C(z^{-1}) = 0$ a $d = 0$. Then it is possible to rewrite (6) to the form of (7).

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) \tag{7}$$

The model is used for the calculation of future output prediction. There are several different methods how to calculate it. One of the simplest ways (using the inverse matrix) is described in this chapter.

The prediction of N steps is possible to write by the set of equations (8).

$$\begin{aligned} y(k+1) &= b_1u(k) + b_2u(k-1) + \dots \\ &\dots + b_{n+1}u(k-n) - a_1y(k) - a_2y(k-1) - \dots - a_{m+1}y(k-m) \\ \\ y(k+2) &= b_1u(k+1) + b_2u(k) + \dots \\ &\dots + b_{n+1}u(k-n+1) - a_1y(k+1) - a_2y(k) - \dots - a_{m+1}y(k-m+1) \\ \\ y(k+3) &= b_1u(k+2) + b_2u(k+1) + \dots \\ &\dots + b_{n+1}u(k-n+2) - a_1y(k+2) - a_2y(k+1) - \dots \\ &\dots - a_{m+1}y(k-m+2) \\ &\vdots \\ y(k+N) &= b_1u(k+N-1) + b_2u(k+N) + \dots \\ &\dots + b_{n+1}u(k+N-n+1) - a_1y(k+N-1) - \\ &\dots - a_2y(k+N) - \dots - a_{m+1}y(k+N-m+1) \end{aligned} \tag{8}$$

In matrix form it is possible to write

$$\mathbf{A} \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N) \end{bmatrix} = \mathbf{B} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{bmatrix} + \tilde{\mathbf{B}} \begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-n) \end{bmatrix} + \tilde{\mathbf{A}} \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-m) \end{bmatrix} \tag{9}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ -a_1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -a_N & -a_{N-1} & \dots & 1 \end{bmatrix}; \quad \dim(\mathbf{A}) = N \times N$$

$$\mathbf{B} = \begin{bmatrix} b_1 & 0 & \dots & 0 \\ b_2 & b_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ b_N & b_{N-1} & \dots & b_1 \end{bmatrix}; \quad \dim(\mathbf{B}) = N \times N$$

$$\tilde{\mathbf{A}} = \begin{bmatrix} a_1 & a_2 & \cdots & a_{m+1} \\ a_2 & \cdots & a_{m+1} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix}; \quad \dim(\tilde{\mathbf{A}}) = N \times (m+1);$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} b_2 & b_3 & \cdots & b_{n+1} \\ b_3 & \cdots & b_{n+1} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix}; \quad \dim(\tilde{\mathbf{B}}) = N \times n$$

Output prediction $y(k+i)$ is possible to calculate by multiplying the equation (9) by the inverse matrix \mathbf{A}^{-1} , equation (10).

$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N) \end{bmatrix} = \mathbf{A}^{-1}\mathbf{B} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{bmatrix} + \mathbf{A}^{-1}\tilde{\mathbf{B}} \begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-n) \end{bmatrix} + \mathbf{A}^{-1}\tilde{\mathbf{A}} \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-m) \end{bmatrix} \quad (10)$$

Because the last two terms describe only the system history, it is possible to put them together to the matrix \mathbf{F} and the vector of historical output and input $\mathbf{h} = [\mathbf{y} \quad \mathbf{u}]^T$. Thus, it is possible to rewrite the equation of prediction to the form of equation (11).

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{F}\mathbf{h} \quad (11)$$

The aim of MPC is to calculate the vector of manipulated variable by minimizing the cost function (12), described in [Baotic, 2006].

$$J = \mathbf{e}_N^T \cdot \mathbf{e}_N + \lambda \cdot \mathbf{u}^T \cdot \mathbf{u} \quad (12)$$

where \mathbf{e} is vector of control errors (length N), \mathbf{u} is vector of manipulated variables (length N) and λ is weighting coefficient.

The cost function can be modified using output prediction (10) and set point vector \mathbf{w} .

$$J = (\mathbf{w} - \mathbf{G}\mathbf{u} - \mathbf{F}\mathbf{h})^T (\mathbf{w} - \mathbf{G}\mathbf{u} - \mathbf{F}\mathbf{h}) + \lambda \cdot \mathbf{u}^T \cdot \mathbf{u} \quad (13)$$

It is possible to calculate the vector of manipulated variable \mathbf{u} analytically using the square norm, equation (14).

$$\mathbf{u} = (\mathbf{G}^T \cdot \mathbf{G} + \lambda \cdot \mathbf{I})^{-1} \cdot \mathbf{G}^T \cdot (\mathbf{w} - \mathbf{F}\mathbf{h}) \quad (14)$$

Only one actual value of the manipulated variable (the first element of the vector) is needed, therefore the final form of the control law is equation (15).

$$u = \mathbf{K} \cdot \begin{bmatrix} w(t) \\ w(t+1) \\ \vdots \\ w(t+N) \end{bmatrix} - \mathbf{F} \cdot \begin{bmatrix} u(t-1) \\ u(t-2) \\ \vdots \\ u(t-n) \\ y(t) \\ y(t-1) \\ \vdots \\ y(t-m) \end{bmatrix} \tag{15}$$

3.2.2 Method modification

GPC theory is formulated in detail for the group of linear systems but in the case of nonlinear systems it is not possible to use it because linear models are not able to describe nonlinear processes well enough. Nonlinear process control needs better description using piecewise linearized model.

In the case of linearized MPC several points where the linearization is done are chosen and for each point controller setting (matrices **G** and **F**) is pre-calculated. Then the controller switches between pre-calculated settings during control experiment (according to actual reactor temperature) and it is possible to interpolate between two adjoining settings. Thus, nonlinear behavior of the system is substituted by piecewise linearized model, more in [Mares et al., 2010b].

Control law can be transformed to equation (16).

$$u = \mathbf{K} \cdot \begin{bmatrix} w(t) \\ w(t+1) \\ \vdots \\ w(t+N) \end{bmatrix} - \bar{\mathbf{F}} \cdot \begin{bmatrix} u(t-1) \\ u(t-2) \\ \vdots \\ u(t-n) \\ y(t) \\ y(t-1) \\ \vdots \\ y(t-m) \end{bmatrix} \tag{16}$$

where vector **K** is the same as in equation (14) and vector $\bar{\mathbf{F}}$ is product of matrices **K** and **F** multiplying. Interpolation is the main reason of multiplying (it is simpler to interpolate between vectors than matrices).

The whole algorithm can be written as:

1. Pre-control
 - fill the data history*
 - calculate vectors T_{LIN} , K_{LIN} and $\bar{\mathbf{F}}_{LIN}$*
2. Control
 - a. *measure actual temperature*
 - b. *choose the interval K_{T_i} $K_{T_{i+1}}$ and $\bar{\mathbf{F}}_{T_i}$ a $\bar{\mathbf{F}}_{T_{i+1}}$*
 - c. *using interpolation calculate vectors **K** a $\bar{\mathbf{F}}$ for the control law*
 - d. *calculate the actual value of manipulated variable *u**
 - e. *actualize the data history*

Vectors T_{LIN} , K_{LIN} and \bar{F}_{LIN} are defined as

$$T_{LIN} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix}; \quad K_{LIN} = \begin{bmatrix} K_{T_1} \\ K_{T_2} \\ \vdots \\ K_{T_N} \end{bmatrix}; \quad \bar{F}_{LIN} = \begin{bmatrix} \bar{F}_{T_1} \\ \bar{F}_{T_2} \\ \vdots \\ \bar{F}_{T_N} \end{bmatrix}$$

3.2.3 Neural network predictive control

There are several variations of neural network predictive controller. This approach uses a neural network model of nonlinear plant to predict future plant performance. The controller then calculates the control input that will optimize plant performance over a specified future time horizon.

The first stage of neural network predictive control is to design a neural network which represents the dynamics of the plant. The prediction error between the plant output and neural network output is used as the neural network training signal [Nguyen et al., 2003]. Obtained neural network predicts the plant response over a specified time horizon. The predictions are used by some search technique to determine the control signal that minimizes the following performance criterion over the specified horizon N

$$J = e_N^T \cdot e_N + \lambda \cdot \Delta u^T \cdot \Delta u \tag{17}$$

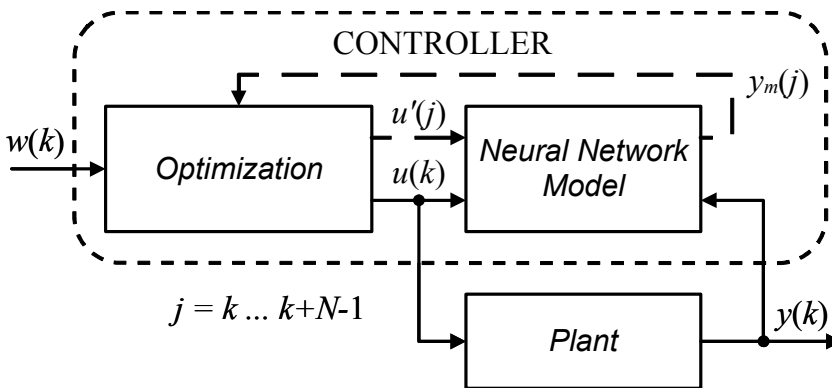


Fig. 3. Neural network predictive control

where e is vector of control errors and Δu is vector of controller outputs differences in time. The following figure illustrates the neural network predictive control process. The controller consists of the neural network plant model and the optimization block. The optimization block determines the values of u' that minimize the criterion J and the optimal u' is input to the plant.

In section 4.4, experimental results of typical Model Predictive Control performance are compared to Neural Network Predictive Control ones.

3.3 Discrete controller tuning online

This technique amplifies the basic feedback control loop. It aims to tune any discrete controller online. For this purpose the knowledge of the controlled system model (e.g. neural model) and

reference variable course over known future finite horizon is necessary. Based on this, the parameters of any chosen discrete controller are determined repeatedly every discrete time instant so that the control response computed via the neural model over future horizon is optimal (according to chosen performance criterion). Simplified scheme is depicted in Fig. 4. The search of discrete controller parameters has to run repeatedly in every single step of the sampling interval, which puts great demands on computing time of the search algorithm. Naturally, usage of some iterative optimization algorithm with only one iteration realization every time instant is suggested. Gradient descent techniques seem inconvenient because of neural model usage. Neural model is black-box-like model so it is not possible to determine gradient descent analytically. On the other hand, evolutionary search techniques (genetic algorithm, differential evolution, ... see [Coello et al., 2002]) appear to be suitable because these techniques do not require any particular information about search problem. The other indisputable advantage is its operating principle. In each iteration, evolutionary search techniques explore not only one value of input variables but whole set of them (one generation of individual solutions), which lowers significantly troubles with initial parameters random choice. In this particular case, differential evolution is chosen. The reasons are, among others, that differential evolution works with decimal input values (contrary to genetic algorithm) and population of possible solutions is kept more diversified.

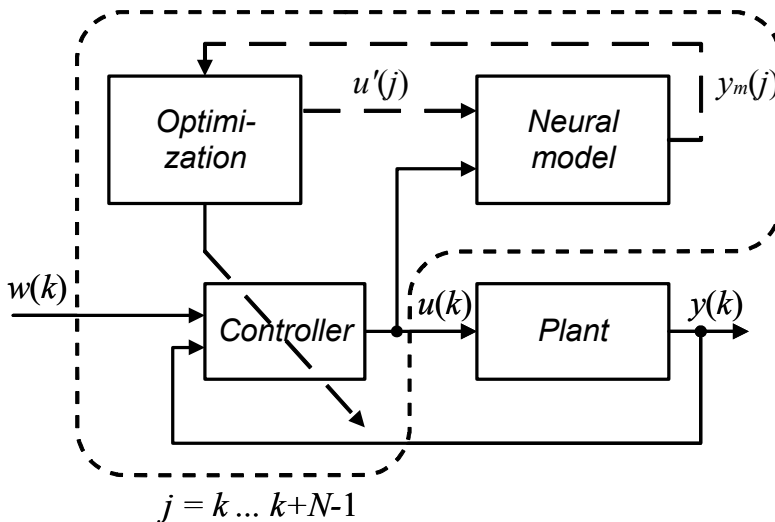


Fig. 4. Controller tuning online using neural network

The control method which is described here does not require any special form of discrete controller. According to some experiments [Dolezel et al., 2009a], [Dolezel et al., 2009b], controller form

$$u(k) = p_0 w(k) - p_1 y(k) - p_2 y(k-1) - p_3 y(k-2) + u(k-1) \tag{18}$$

was considered to be convenient. For some $p_0 \dots p_3$ parameters combinations, controller (18) acts like discrete PID controller. In general, however, it has one additional independent parameter. Suitable control performances can be obtained by well-tuned controller (18). Because of the evolutionary algorithm, cost function can be selected from huge number of possibilities. One of suitable definitions is

$$J = \frac{1}{N} \cdot \sum_{i=k}^{k+N-1} |e(i)| + \frac{h_1}{N-1} \cdot \sum_{i=k+1}^{k+N-1} |\Delta u(i)| + h_2 \cdot |e(k+N-1)| \tag{19}$$

where $\Delta u(i) = u(i) - u(i-1)$, $e(i)$ is control error, h_1 is function parameter influencing manipulated variable differences, h_2 is function parameter influencing the state on the end of future horizon, N is future horizon length and $w(i)$ is reference variable.

Note that definition (19) can be changed in order to get any particular control performance. The whole algorithm of the above described control method is compiled in the following points:

1. Create dynamical neural model of controlled system – see [Nguyen, 2003]
2. Choose future horizon length N
3. Choose differential evolution parameters (number of individual solutions in one generation NP – any solution represents one particular quaternion of controller parameters $p_0 \dots p_3$, crossover constant CR , mutation constant F) and their initial values
4. Measure controlled variable $y(k)$
5. Perform one iteration of differential evolution (based on the knowledge of controlled variable $y(k)$, course of its reference $w(k)$ till $w(k+N-1)$ and neural model of controlled system)
 - a. perform control simulation with discrete controller and the neural model over future horizon N and evaluate cost function for all the individual solutions from current generation
 - b. Apply cross-over and mutation (see [Coello et al., 2002]) so that offspring generation of solutions is bred
 - c. Evaluate cost functions of offspring (see step a)
 - d. Choose the best individual solution from the offspring generation
6. Evaluate manipulated variable $u(k)$ with discrete controller determined by the best individual solution obtained in step 5d)
7. $k = k + 1$, go to step 4

4. Experimental results

4.1 Dynamical neural model of the plant

Control techniques described above need neural plant model to be designed. In [Nguyen, 2003], whole algorithm of neural model design is presented in detail. First, a training set of process data is to be measure. For this purpose, a simple control experiment with reactor furnace and PI controller is performed – see Fig. 5.

Data (sampling interval 3s) are slotted according to Table 1 so that neural network corresponds to difference equation (20)

$$T_D^M(k) = \varphi[T_D^M(k-1), T_D^M(k-2), E(k-1), E(k-2)] \tag{20}$$

Input				Output
$T_D^M(k-1)$	$T_D^M(k-2)$	$E(k-1)$	$E(k-2)$	$T_D^M(k)$
$T_D^M(2)$	$T_D^M(1)$	$E(2)$	$E(1)$	$T_D^M(3)$
$T_D^M(3)$	$T_D^M(2)$	$E(3)$	$E(2)$	$T_D^M(4)$
...
$T_D^M(N-1)$	$T_D^M(N-2)$	$E(N-1)$	$E(N-2)$	$T_D^M(N)$

Table 1. Training set 1

It is possible to choose higher order of difference equation (20), but after many experiments, second order seems convenient. Formal scheme of the neural model can be found in Fig. 6.

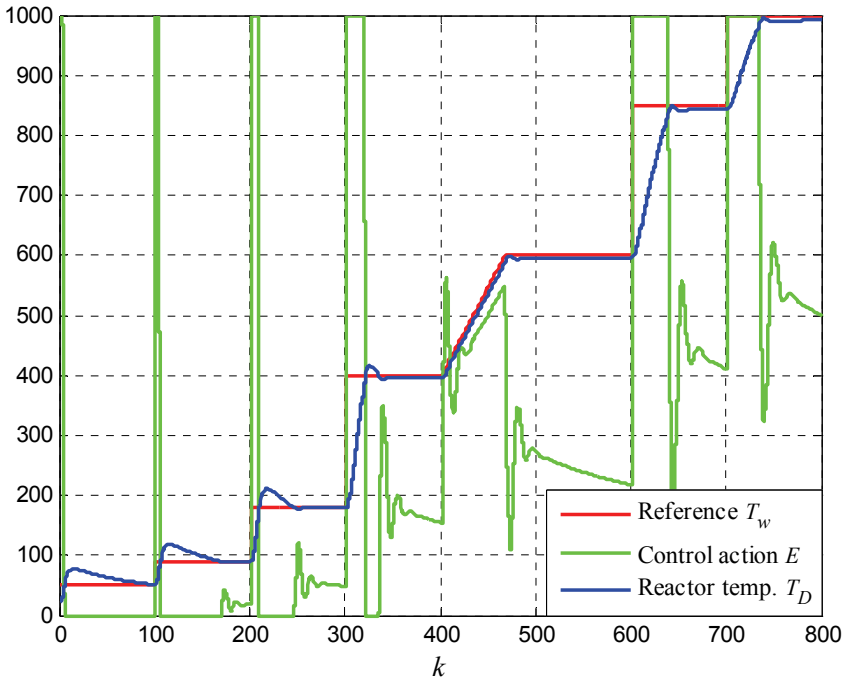


Fig. 5. Control performance with PI controller – Training set experiment

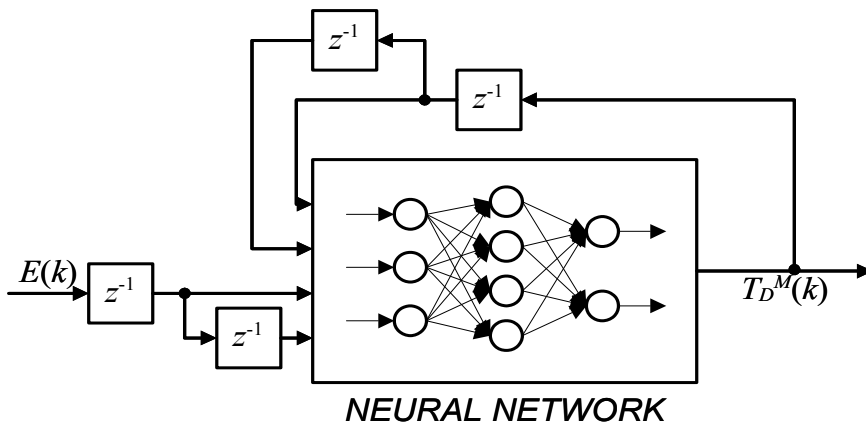


Fig. 6. Formal scheme of the neural model

Now, topology of the neural network has to be optimized. Several neural networks with different number of hidden neurons were trained (Levenberg-Marquardt Algorithm was used) and cost function courses are depicted in Fig. 7.

For control experiments neural model with network of 4-6-1 topology is chosen, because networks with more complex topologies do not bring considerably improved performances.

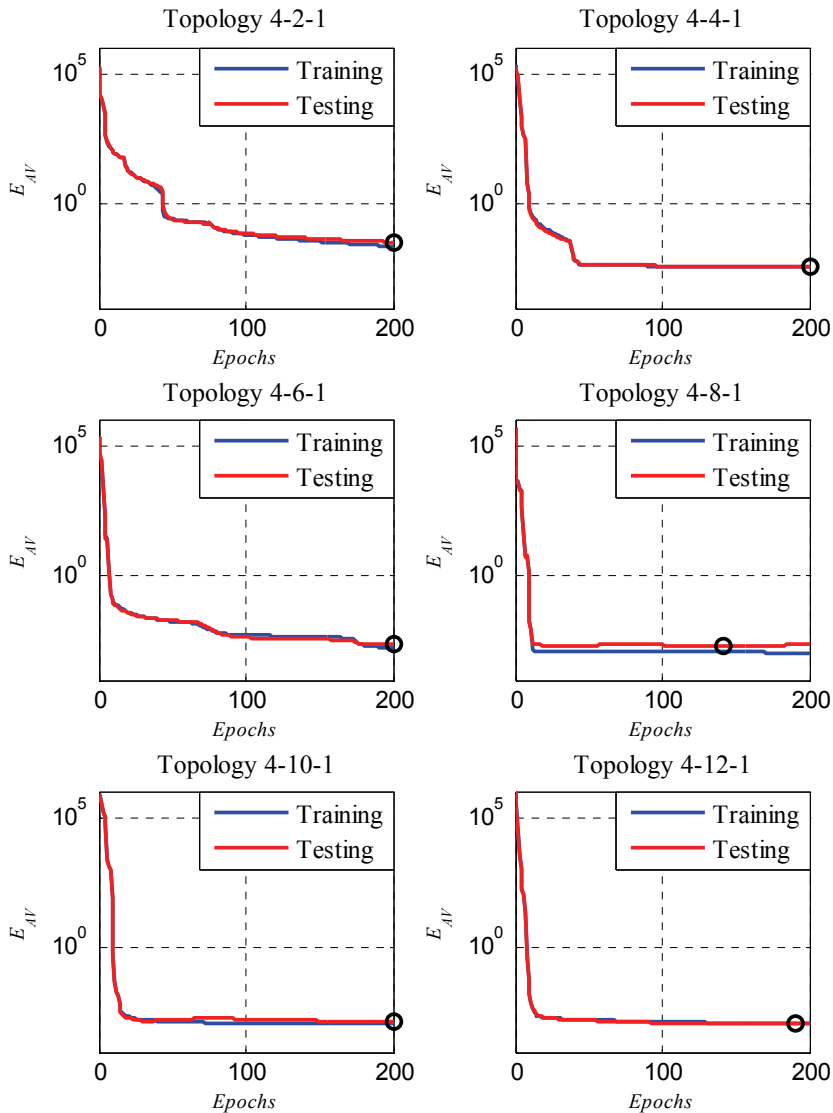


Fig. 7. Topology optimizing I

4.2 Inverse neural model of the plant

For IMC control an inverse plant model is needed. Inverse neural model control design is formally the same as feedforward model, the only difference is, that data of training set has to be slotted in another way. Inverse difference equation of equation (20) can be obtained by actual input-actual output interchanging – equation (21).

$$E(k-1) = \varphi[T_D^M(k), T_D^M(k-1), T_D^M(k-2), E(k-2)] \quad (21)$$

Input				Output
$TDM(k)$	$TDM(k-1)$	$TDM(k-2)$	$E(k-2)$	$E(k-1)$
$T_D^M(3)$	$T_D^M(2)$	$T_D^M(1)$	$E(1)$	$E(2)$
$T_D^M(4)$	$T_D^M(3)$	$T_D^M(2)$	$E(2)$	$E(3)$
...
$T_D^M(N)$	$T_D^M(N-1)$	$T_D^M(N-2)$	$E(N-2)$	$E(N-1)$

Table 2. Training set 2

Equation above lacks time causality. However, it can be used to training set slotting - see Table 2.

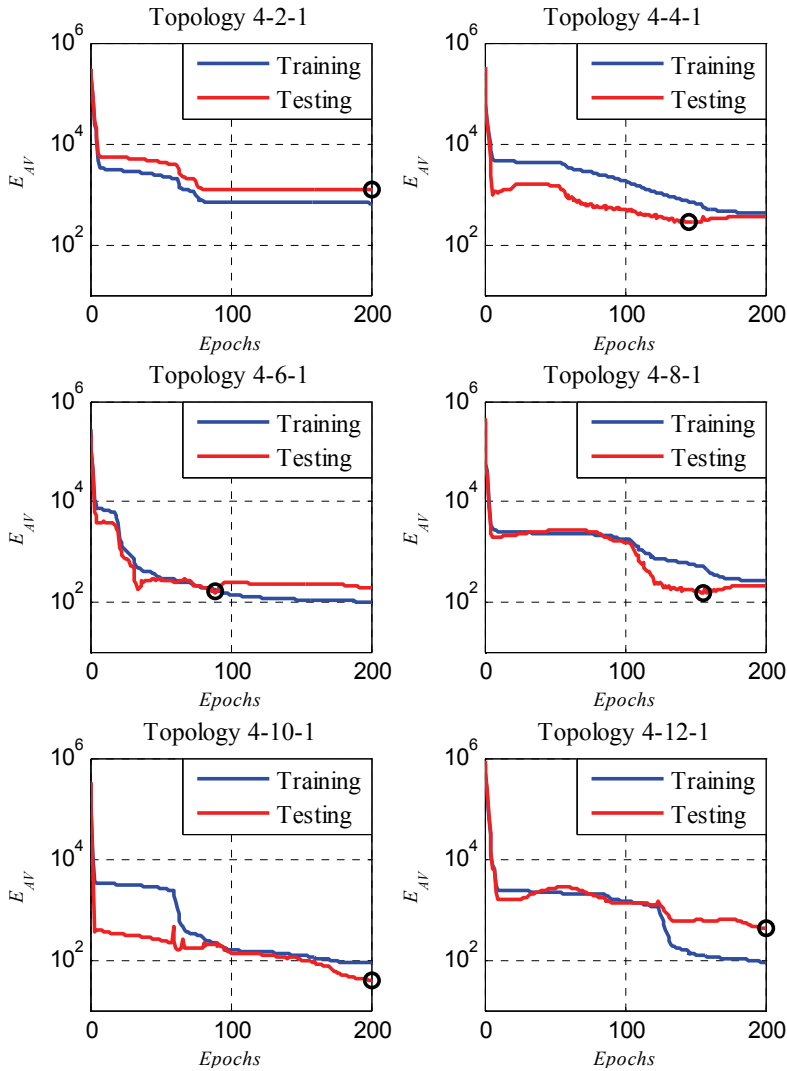


Fig. 8. Topology optimizing II

Topology is optimized as well as in section 4.1. Cost function courses are depicted in Fig. 8. Now, inverse neural model with 4-10-1 topology is chosen.

4.3 Neural internal model control

If both feedforward and inverse neural models are designed, control loop can be put together – see Fig. 9.

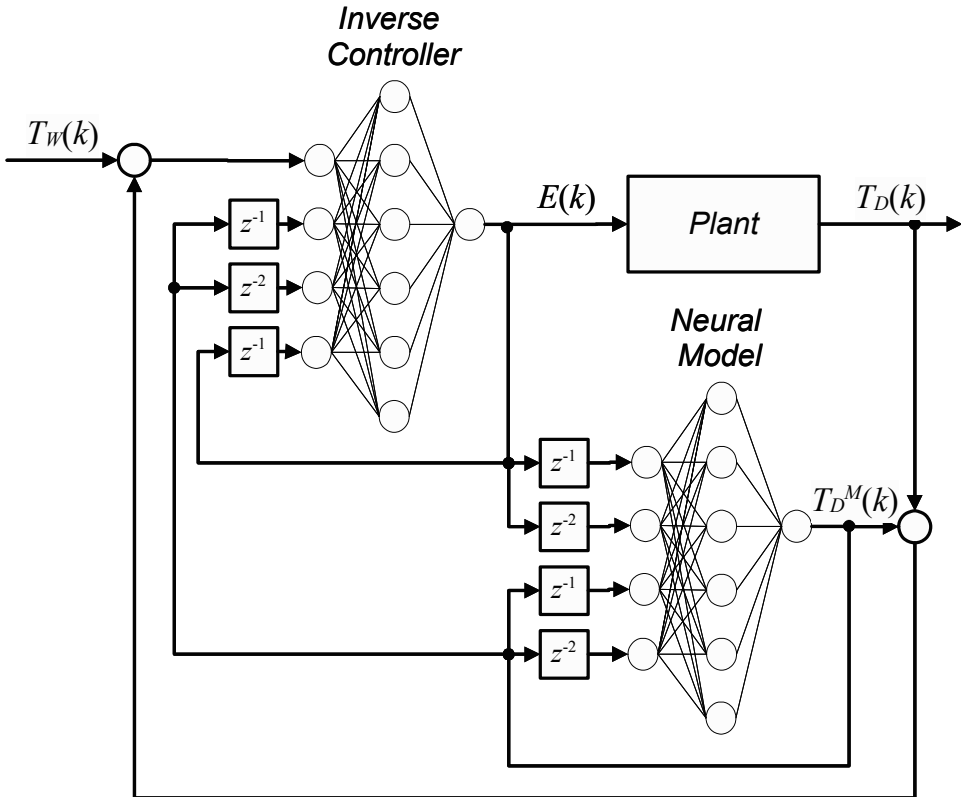


Fig. 9. Reactor Furnace controlled using Neural IMC

Control response for the desired plant output course is shown below compared to response obtained by classical IMC (for linear model design, same data is used).

For ramp as reference, reactor temperature courses are similar for both control techniques (in addition, control response with classical IMC is smoother), so for reactor furnace control, it is no need to extend classical IMC technique with neural networks.

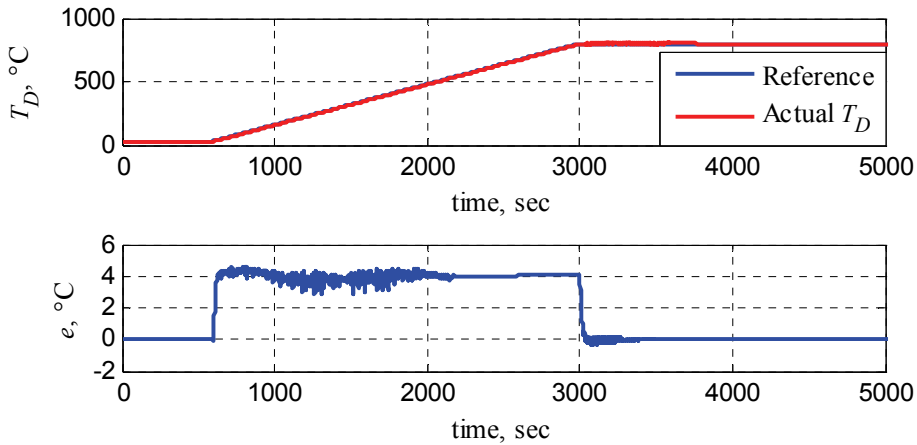


Fig. 10. Neural IMC control response

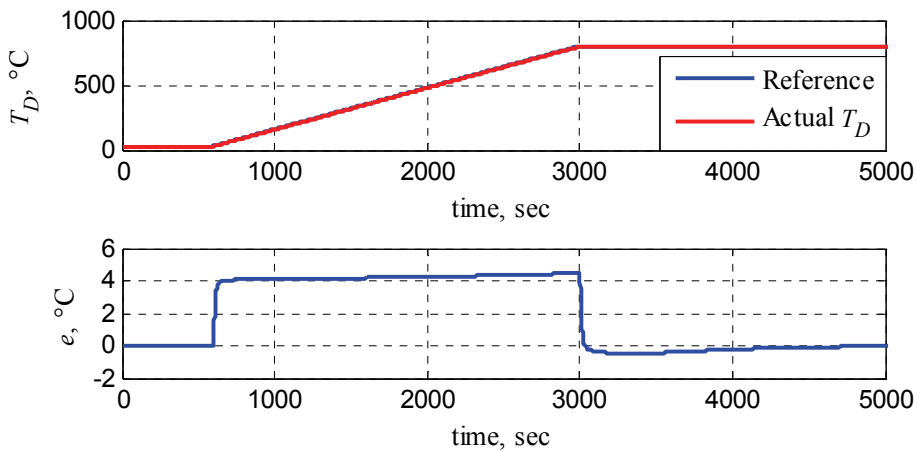


Fig. 11. Classical IMC control response

4.4 Neural network predictive control

As shown in section 3.2.3, only feedforward neural model of the plant is needed for neural network predictive control. Control experiment is performed according to section 3.2.3 with golden section search routine [Fletcher, 1987], prediction horizon $N = 20$ and weighting coefficient $\lambda = 0.1$. Control response is shown in Fig. 12.

Alternatively, control response gained by piecewise linearized model predictive controller (the same prediction horizon and weighting coefficient - see section 3.2.1 and 3.2.2) is plotted in Fig. 13. It is obvious that neural network predictive controller provides less suitable controlperformance. However, it has to be mentioned, that neural network predictive controller is much simpler to design than piecewise linear model predictive controller.

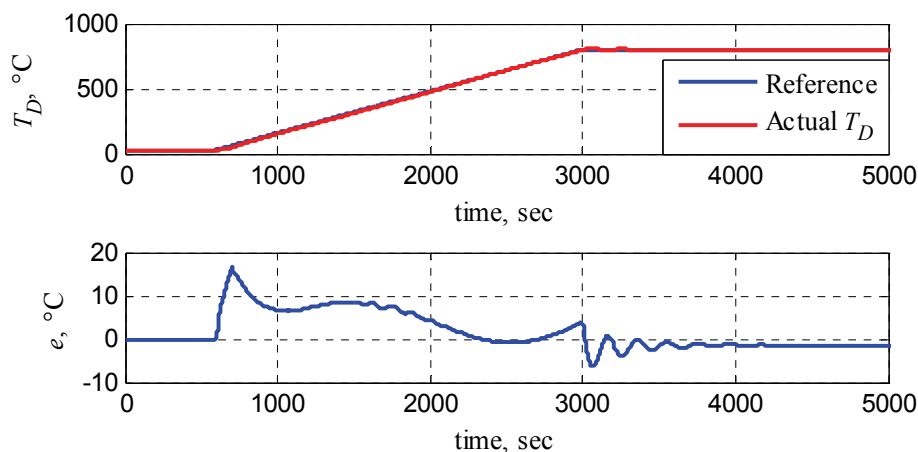


Fig. 12. Neural Network Predictive control response

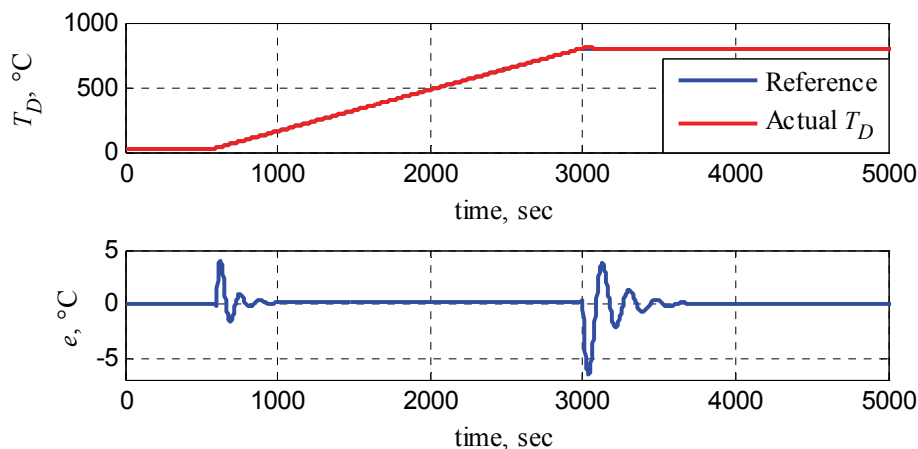


Fig. 13. Piecewise Linearized Model Predictive control response

4.5 Discrete controller tuning online

Control loop of this technique is connected in a way introduced briefly in section 3.3. Differential evolution is chosen as search technique. After some experiments, eligible parameters are chosen this way: $NP = 30$; $CR = 0.85$; $F = 0.6$; $N = 20$. Cost function is selected according to Eq. (19), where $h_1 = 0.1$, $h_2 = 0.01$. Control response is depicted in Fig. 14.

There is no exact alternative in classical control theory to this technique. However, in a certain way it is close to predictive control, therefore it can be compared to Fig. 13.

It is remarkable, that control response shown in Fig. 14 provides the most suitable performance of all experiments. But, on the other hand, it is highly computationally demanding technique.

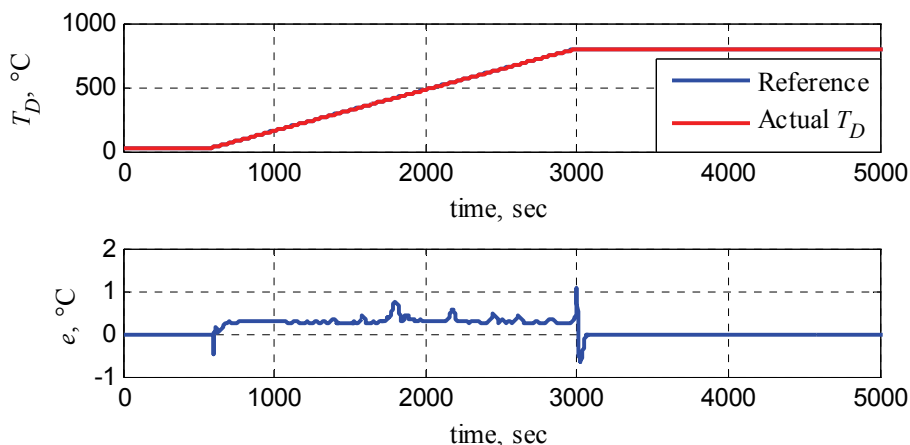


Fig. 14. Discrete controller tuned online

5. Conclusion

The aim of this work was to design a controller, which provides control performance with control error less than 10°C . Because of the nonlinearity of the plant, two groups of advanced control techniques were used. The first group is based on artificial neural networks usage while the second one combines their alternatives in modern control theory. Generally speaking, neural networks are recommended to use when plant is strongly nonlinear and/or stochastic. Although reactor furnace is indispensably nonlinear, it is evident that control techniques without neural networks can control the plant sufficiently and in some cases (especially predictive control and internal model control) even better. Thus, neural network usage is not strictly necessary here, although especially Discrete Controller Tuning Online brings extra good performance.

6. Acknowledgement

This work was supported by the 6th Framework Programme of the European Community under contract No. NMP2-CT-2007-026515 "Bioproduction project - Sustainable Microbial and Biocatalytic Production of Advanced Functional Materials" and by the funds No. MSM 6046137306 and No. MSM 0021627505 of the Czech Ministry of Education. This support is very gratefully acknowledged.

7. References

- Baotic, M.; Christophensen, F.; Morari, M. (2006). Constrained Optimal Control of Hybrid Systems With a Linear Performance Index. *IEEE Trans. on Automatic Control*, Vol.51, No 12., ISSN 1903-1919.
- Camacho, E.F.; Bordons, C. (2007). *Model Predictive Control*, Springer-Verlag, ISBN 1-85233-694-3, London

- Coello, C. A. C.; Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-Objective problems*, Springer, ISBN 978-0-387-33254-3, Boston
- Dolezel, P.; Taufer, I. (2009a). PSD controller tuning using artificial intelligence techniques, *Proceedings of the 17th International Conference on Process Control '09*, pp. 120-124, ISBN 978-80-227-3081-5, Strbske Pleso, June 2009, STU, Bratislava.
- Dolezel, P.; Mares, J. (2009b). Reactor Furnace Control using Artificial Neural Networks and Genetic Algorithm, *Proceedings of the International Conference on Applied Electronics*, pp. 99-102, ISBN 978-80-7043-781-0, Plzen, September 2009, ZCU, Plzen.
- Dwarapudi, S.; Gupta, P. K.; Rao, S. M. (2007). Prediction of iron ore pellet strength using artificial neural network model, *ISIJ International*, Vol. 47, No 1., ISSN 0915-1559.
- Economou, C.; Morari, M.; Palsson, B. (1986). Internal Model Control: extension to nonlinear system, *Industrial & Engineering Chemistry Process Design and Development*, Vol. 26, No 1, pp. 403-411, ISSN 0196-4305.
- Fletcher, R. (1987). *Practical Methods of Optimization*, Wiley, ISBN 978-0-471-91547-8, Chichester, UK.
- Lichota, J.; Grabovski, M. (2010). Application of artificial neural network to boiler and turbine control, *Rynek Energii*, ISSN 1425-5960.
- Mares, J., Dusek, F., Dolezel, P. (2010a) Nelinearni a linearizovany model reaktorové pece. In *Proceedings of Conference ARTEP'10*, 24.-26. 2 2010. Technická univerzita Kosice, 2010. Pp. 27-1 - 27-14. ISBN 978-80-553-0347-5.
- Mares, J., Dusek, F., Dolezel, P. (2010b). Prediktivni rizeni reaktorove pece. In *Proceedings of XXXVth Seminary ASR'10 „Instruments and Control”*, VSB- Technical University Ostrava, 2010. Pp. 269 - 279. ISBN 978-80-248-2191-7.
- Montague, G.; Morris, J. (1994). Neural network contributions in biotechnology, *Trends in biotechnology*, Vol. 12, No 8., ISSN 0167-7799.
- Nguyen, H.; Prasad, N.; Walker, C. (2003). *A First Course in Fuzzy and Neural Control*, Chapman & Hall/CRC, ISBN 1-58488-244-1, Boca Raton.
- Norgaard, M.; Ravn, O.; Poulsen, N. (2000). *A Neural Networks for Modelling and Control of Dynamic Systems*, Springer-Verlag, ISBN 978-1-85233-227-3, London.
- Rivera, D.; Morari, M.; Skogestad, S. (1986). Internal Model Control: PID Controller Design, *Industrial & Engineering Chemistry Process Design and Development*, Vol. 25, No 1., pp. 252-265, ISSN 0196-4305.
- Teixeira, A.; Alves, C.; Alves, P. M. (2005). Hybrid metabolic flux analysis/artificial neural network modelling of bioprocesses, *Proceedings of the 5th International Conference on Hybrid Intelligent Systems*, ISBN 0-7695-2457-5, Rio de Janeiro.



Artificial Neural Networks - Industrial and Control Engineering Applications

Edited by Prof. Kenji Suzuki

ISBN 978-953-307-220-3

Hard cover, 478 pages

Publisher InTech

Published online 04, April, 2011

Published in print edition April, 2011

Artificial neural networks may probably be the single most successful technology in the last two decades which has been widely used in a large variety of applications. The purpose of this book is to provide recent advances of artificial neural networks in industrial and control engineering applications. The book begins with a review of applications of artificial neural networks in textile industries. Particular applications in textile industries follow. Parts continue with applications in materials science and industry such as material identification, and estimation of material property and state, food industry such as meat, electric and power industry such as batteries and power systems, mechanical engineering such as engines and machines, and control and robotic engineering such as system control and identification, fault diagnosis systems, and robot manipulation. Thus, this book will be a fundamental source of recent advances and applications of artificial neural networks in industrial and control engineering areas. The target audience includes professors and students in engineering schools, and researchers and engineers in industries.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jan Mareš, Petr Doležel and Pavel Hrnčířik (2011). Artificial Neural Network – Possible Approach to Nonlinear System Control, *Artificial Neural Networks - Industrial and Control Engineering Applications*, Prof. Kenji Suzuki (Ed.), ISBN: 978-953-307-220-3, InTech, Available from: <http://www.intechopen.com/books/artificial-neural-networks-industrial-and-control-engineering-applications/artificial-neural-network-possible-approach-to-nonlinear-system-control>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.