Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation, and Mathematics

PROCEEDINGS

of the 18$^{\text{th}}$ International Conference on Process Control

Hotel Titris, Tatranská Lomnica, Slovakia, June 14 – 17, 2011

ISBN 978-80-227-3517-9

http://www.kirp.chtf.stuba.sk/pc11

Editors: M. Fikar and M. Kvasnica

Doležel, P., Taufer, I., Mareš, J.: Piecewise-Linear Neural Models for Process Control, Editors: Fikar, M., Kvasnica, M., In *Proceedings of the 18th International Conference on Process Control*, Tatranská Lomnica, Slovakia, 296–300, 2011.

Full paper online: http://www.kirp.chtf.stuba.sk/pc11/data/abstracts/043.html

Piecewise-Linear Neural Models for Process Control

P. Doležel* I. Taufer J. Mareš*****

** University of Pardubice, Faculty of Electrical Engineering and Informatics, Department of Process Control, Nám. Čs. legií 565, 532 10 Pardubice*
(Tel: +420 466 037 504; e-mail:petr.dolezel@upce.cz.)
*** University of Pardubice, Faculty of Electrical Engineering and Informatics, Department of Process Control, Nám. Čs. legií 565, 532 10 Pardubice*
(e-mail:ivan.taufer@upce.cz.)
**** Institute of Chemical Technology, Department of Computing and Control Engineering, Technická 5, 166 28 Prague 6*
(e-mail: jan.mares@vscht.cz)

Abstract: There is introduced an algorithm which provides piecewise-linear model of nonlinear plant using artificial neural networks, in this paper. That piecewise-linear model is precise and each linear submodel is valid in some neighbourhood of actual plant state. This model can be used for plant control design. There is presented an example at the end of this paper, where defined nonlinear plant is controlled via Pole Assignment technique using piecewise-linear neural model and control response is compared to data obtained by common PID controller.

1. INTRODUCTION

Artificial Neural Network (ANN) is a popular methodology nowadays with lots of practical and industrial applications. As introduction it is necessary to mention applications as mathematical modelling of bioprocesses in Montague et al. (1994), Teixeira et al. (2005), prediction models and control of boilers, furnaces and turbines in Lichota et al. (2010) or industrial ANN control of calcinations processes and iron ore processes in Dwarapudi, et al. (2007).

Therefore, the aim of the contribution is to explain how to use ANN with piecewise-linear activation functions in hidden layer in process control. To be more specific, there is described technique of controlled plant linearization using ANN nonlinear model. Obtained linearized model is in a shape of linear difference equation.

2. ANN FOR APPROXIMATION

According to Kolmogorov's Superposition Theorem, any real continuous multidimensional function can be evaluated by sum of real continuous one-dimensional functions, see Hecht-Nielsen (1987). If the theorem is applied to ANN, it can be said that any real continuous multidimensional function can be approximated by certain three-layered ANN with arbitrary precision. Topology of that ANN is depictured in Fig. 1. Input layer brings external inputs x_1, x_2, …, x_P into ANN. Hidden layer contains S neurons, which process sums of weighted inputs using continuous, bounded and monotonic activation function. Output layer contains one neuron, which processes sum of weighted outputs from hidden neurons. Its activation function has to be continuous and monotonic.

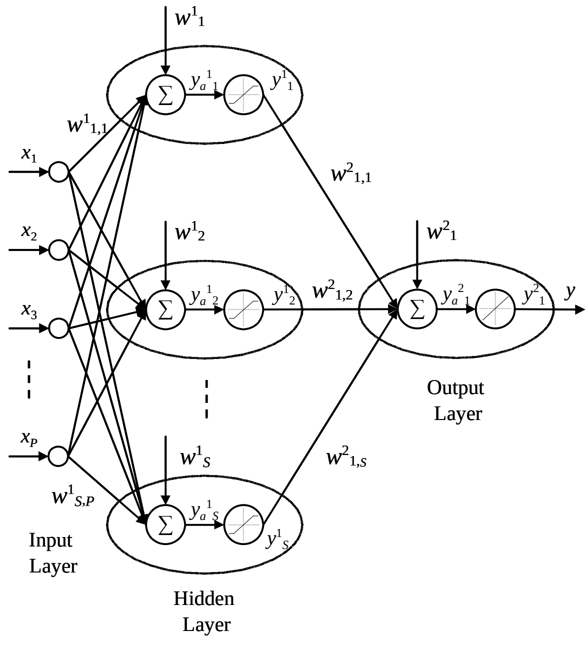

Fig. 1. Three-layered ANN

So ANN in Fig. 1 takes P inputs, those inputs are processed by S neurons in hidden layer and then by one output neuron. Dataflow between input i and hidden neuron j is gained by weight $w^1_{j,i}$. Dataflow between hidden neuron k and output neuron is gained by weight $w^2_{1,k}$. Output of the network can be expressed by following equations.

$$y_a{}^1{}_j = \sum_{i=1}^{P} w^1{}_{j,i} \cdot x_i + w^1{}_j \tag{1}$$

$$y^1{}_j = \varphi^1\left(y_a{}^1{}_j\right) \tag{2}$$

$$y_a{}^2{}_1 = \sum_{i=1}^{S} w^2{}_{1,i} \cdot y^1{}_i + w^2{}_1 \quad (3)$$

$$y = \varphi^2\left(y_a{}^2{}_1\right) \quad (4)$$

In equations above, $\varphi^1(.)$ means activation functions of hidden neurons and $\varphi^2(.)$ means output neuron activation function.

As it is mentioned above, there are some conditions applicable for activation functions. To satisfy those conditions, there is used mostly hyperbolic tangent activation function (eq. 5) for neurons in hidden layer and identical activation function (eq. 6) for output neuron.

$$y^1{}_j = \tanh\left(y_a{}^1{}_j\right) \quad (5)$$

$$y = y_a{}^2{}_1 \quad (6)$$

Mentioned theorem does not define how to set number of hidden neurons or how to tune weights. However, there have been published many papers which are focused especially on gradient training methods (Back-Propagation Gradient Descend Alg.) or derived methods (Levenberg-Marquardt Alg.) – see Haykin (1994).

3. SYSTEM IDENTIFICATION BY ANN

System identification means especially a procedure which leads to dynamic model of the system. ANN has traditionally enjoyed considerable attention in system identification because of its outstanding approximation qualities. There are several ways to use ANN for system identification. One of them assumes that the system to be identified (with input u and output y_S) is determined by the following nonlinear discrete-time difference equation.

$$y_S(k) = \psi[y_S(k-1), \ldots, y_S(k-n), \\ u(k-1), \ldots, u(k-m)], m \le n \quad (7)$$

In equation above, $\psi(.)$ is nonlinear function, k is discrete time and n is difference equation order.

The aim of the identification is to design ANN which approximates nonlinear function $\psi(.)$. Then, neural model can be expressed by (eq. 8).

$$y_M(k) = \hat{\psi}[y_M(k-1), \ldots, y_M(k-n), \\ u(k-1), \ldots, u(k-m)], m \le n \quad (8)$$

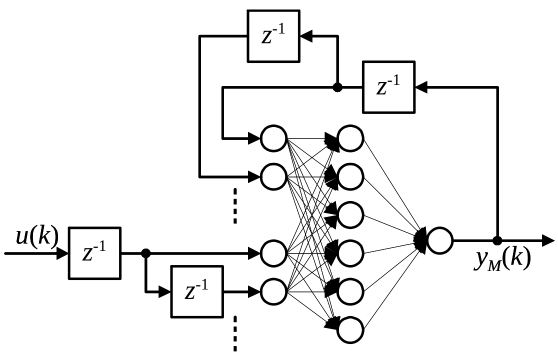

Fig. 2. Neural model

In (eq. 8), $\hat{\psi}$ represents well trained ANN and y_M is its output. Formal scheme of neural model is shown in Fig. 2. It is obvious that ANN in Fig. 2 has to be trained to provide y_M as close to y_S as possible. Existence of such a neural network is guaranteed by Kolmogorov's Superposition Theorem and whole process of neural model design is described in detail in Haykin (1994) or Taufer et al. (2008).

4. PIECEWISE-LINEAR MODEL

As mentioned in section 2, there is recommended to use hyperbolic tangent activation function for neurons in hidden layer and identical activation function for output neuron in ANN used in neural model. However, if linear saturated activation function (eq. 9) is used instead, ANN features stay similar because of resembling courses of both activation functions (see Fig. 3).

$$y^1{}_j = \begin{cases} 1 & \text{for} \quad y_a{}^1{}_j > 1 \\ y_a{}^1{}_j & \text{for} \quad -1 \le y_a{}^1{}_j \le 1 \\ -1 & \text{for} \quad y_a{}^1{}_j < -1 \end{cases} \quad (9)$$

The output of linear saturated activation function is either constant or equal to input so neural model which uses ANN with linear saturated activation functions in hidden neurons acts as piecewise-linear model. One linear submodel turns to another when any hidden neuron becomes saturated or becomes not saturated.

Let us presume an existence of some dynamic neural model which uses ANN with linear saturated activation functions in hidden neurons and identic activation function in output neuron – see Fig. 4. Let us also presume $m = n = 2$ for making process easier. ANN output can be computed using eqs. (1), (2), (3), (4). However, another way for ANN output computing is useful. Let us define saturation vector $\mathbf{z}$ of S elements. This vector indicates saturation states of hidden neurons – see (eq. 10).

$$z_i = \begin{cases} 1 & \text{for} \quad y^1{}_i > 1 \\ 0 & \text{for} \quad -1 \le y^1{}_i \le 1 \\ -1 & \text{for} \quad y^1{}_i < -1 \end{cases} \quad (10)$$

Then, ANN output can be expressed by (eq. 11).

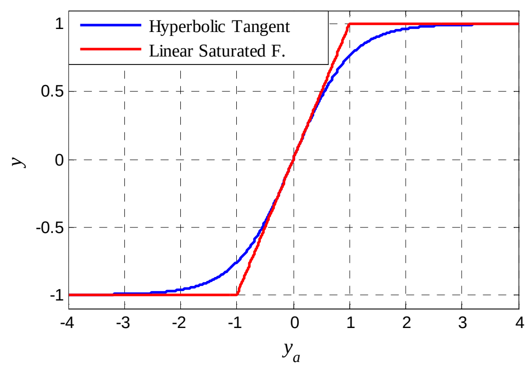

Fig. 3. Activation functions comparison

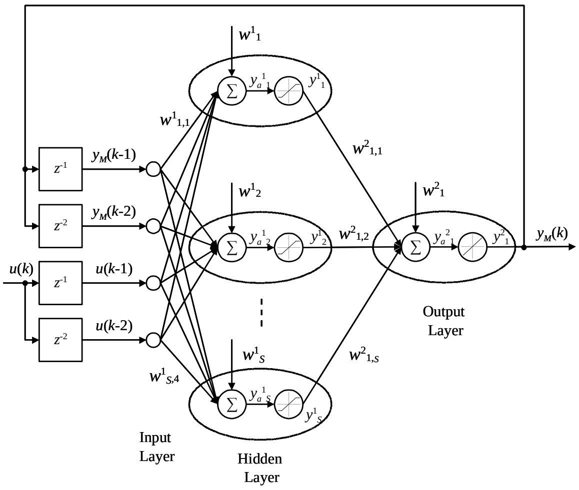

Fig. 4. Piecewise-linear neural model

$$y_M(k) = -a_1 \cdot y_M(k-1) - a_2 \cdot y_M(k-2) + \\ + b_1 \cdot u(k-1) + b_2 \cdot u(k-2) + c \tag{11}$$

where:

$$a_1 = -\sum_{i=1}^{S} w^2_{1,i} \cdot \left(1 - |z_i|\right) \cdot w^1_{i,1}$$

$$a_2 = -\sum_{i=1}^{S} w^2_{1,i} \cdot \left(1 - |z_i|\right) \cdot w^1_{i,2}$$

$$b_1 = \sum_{i=1}^{S} w^2_{1,i} \cdot \left(1 - |z_i|\right) \cdot w^1_{i,3}$$

$$b_2 = \sum_{i=1}^{S} w^2_{1,i} \cdot \left(1 - |z_i|\right) \cdot w^1_{i,4}$$

$$c = w^2_1 + \sum_{i=1}^{S} \left(w^2_{1,i} \cdot z_i + \left(1 - |z_i|\right) \cdot w^2_{1,i} \cdot w^1_i \right)$$

Thus, difference equation (11) defines ANN output and it is linear in some neighbourhood of actual state (in that neighbourhood, where saturation vector **z** stays constant). Difference equation (11) can be clearly extended into any order.

In other words, if it is designed neural model of any nonlinear system in form of Fig. 4, then it is simple to determine parameters of linear difference equation which approximates system behaviour in some neighbourhood of actual state. This difference equation can be used then to the actual control action setting due to any of classical or modern control techniques.

If chosen control technique requires model in form of difference equation with no constant term (c = 0), (eq. 11) can be transformed in following way. Let us define

$$\tilde{u}(k) = u(k) - u_0 \tag{12}$$

where u_0 is constant. Then, (eq. 11) turns into

$$y_M(k) = -a_1 \cdot y_M(k-1) - a_2 \cdot y_M(k-2) + \\ + b_1 \cdot \tilde{u}(k-1) + b_2 \cdot \tilde{u}(k-2) + c + \\ + (b_1 + b_2) \cdot u_0 \tag{13}$$

Equation (13) becomes constant term free, if (eq. 14) will be satisfied.

$$u_0 = -\frac{c}{b_1 + b_2} \tag{14}$$

It is obvious that mentioned procedure can be extended into any order of difference equation.

Whole algorithm of piecewise-linear neural model usage in process control is summarized in following terms.

1. Create neural model of controlled plant in form of Fig. 4.
2. Set k = 0.
3. Measure plant output $y_S(k)$.
4. Determine parameters a_i, b_i and c of difference equation (11).
5. Transform (eq. 11) into (eq. 13).
6. Determine $\tilde{u}(k)$ according to some chosen control technique using linear plant model in form (eq. 13).
7. Transform $\tilde{u}(k)$ into $u(k)$ using (eq. 12) and perform control action.
8. $k = k + 1$, go to 3.

5. EXAMPLE

Demonstrative nonlinear controlled system is defined by difference equation (15).

$$y_S(k) = \\ \frac{1.5 \cdot y_S(k-1) - 0.8 \cdot y_S(k-2) + 0.1 \cdot u(k-1) + 0.05 \cdot u(k-2)}{1 - 0.1 \cdot y_S(k-1) + 0.2 \cdot [y_S(k-1)]^2} + \\ + 0.6 \cdot \sqrt{u(k-1)} \tag{15}$$

There are defined the boundaries of input $u(k)$ to interval <0;3>. Static characteristic of the system is figured below (Fig. 5).

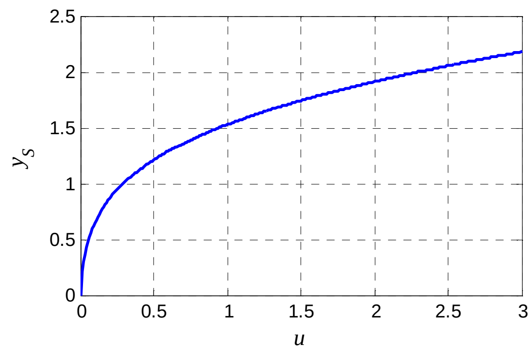

Fig. 5. Static characteristic of the system

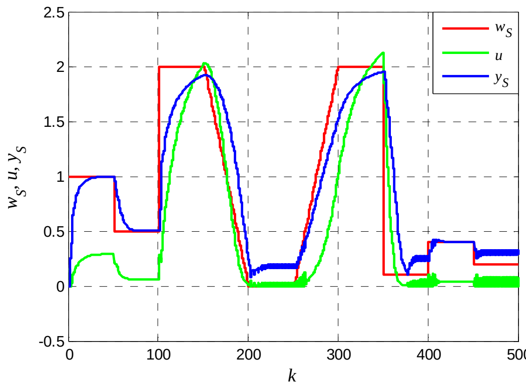

Fig. 6. Control response with PID controller

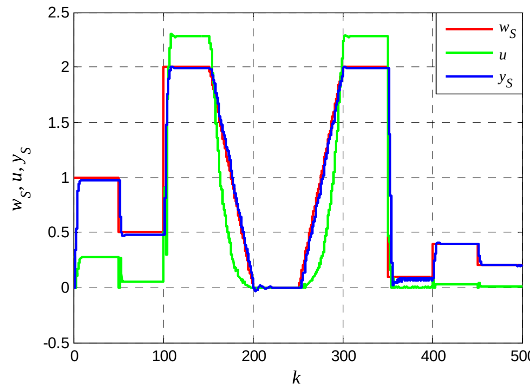

Fig. 8. Control Response with PA Controller and Piecewise-Linear Neural Model

Firstly, system is controlled with PID controller tuned by trial and error – more sophisticated tuning methods fail to bring better performances because of significant nonlinearity of the plant. Control response (Fig. 6) shows serious lack of quality. For lower values of controlled variable $y_S(k)$, control performance oscillates unacceptably, while for higher values of $y_S(k)$, control performance is too damped.

Then, piecewise-linear neural model is used for control. Neural model is designed according to information described in section 4. Detailed description of the process is not referred here, because it is standard well-known procedure. Certain control technique, which can use system model in form of (eq. 13), has to be determined. In this demonstration, Pole Assignment control technique (PA) of Algebraic Control Theory is used.

In simple words, this control technique determines controller parameters so that whole closed control loop behaves as some defined standard. In one its version, PA uses control loop shown in Fig. 7. Controlled system should be described by polynomials $A(z^{-1})$, $B(z^{-1})$, where polynomial parameters are equal to difference equation parameters used for linear model of the controlled system. Both feedforward and feedback part of controller are defined by polynomials $P(z^{-1})$, $Q(z^{-1})$, $R(z^{-1})$, which can be determined by solving of several diophantine equations. Standard for control loop behaviour has to be chosen. Whole procedure of PA is described in detail in book edited by K. J. Hunt (1993).

Standard for this demonstration is defined as discrete first order system with unit gain and denominator $(1 - 0.6065z^{-1})$.

Control performance is shown in Fig. 8. Compared to Fig. 6, there comes clear improvement.

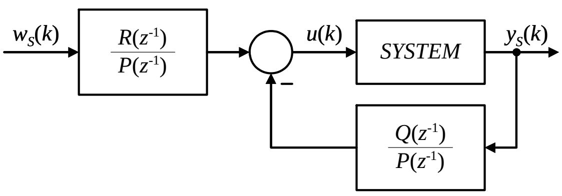

Fig. 7. Pole Assignment Control Technique

6. CONCLUSIONS

The paper is focused on usage of neural network with linear saturated activation functions in process control. Neural model with such a neural network within is suitable for controller design using any of huge set of classical or modern control techniques. As example, there is presented control of nonlinear discrete plant using Pole Assignment technique. Comparison to control performance provided by PID controller proves great improvement.

ACKNOWLEDGMENTS

The work has been supported by the funds No. MSM 6046137306 and No. MSM 0021627505 of Ministry of Education of the Czech Republic, No. MEB 0810003 of Ministry of Education, Science, Research and Sport of the Slovak Republic and of Ministry of Education of the Czech Republic and No. SGFEI06/2011. This support is very gratefully acknowledged.

REFERENCES

Dwarapudi, S., Gupta, P. K. and Rao, S. M. (2007). Prediction of iron ore pellet strength using artificial neural network model, *ISIJ International*, Vol. 47, No 1. pp. 67-72, ISSN 0915-1559.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. Prentice Hall. New Jersey. ISBN 0023527617

Hecht-Nielsen, R. (1987). Kolmogorov’s mapping neural network existence theorem. In: *Proc 1987 IEEE International Conference on Neural Networks.* Vol. 3, pp. 11-13. IEEE Press.

Hunt, K. J., Ed. (1993). *Polynomial methods in optimal control and filtering*. Peter Peregrinus Ltd. Stevenage. ISBN 0-86341-295-5.

Lichota, J. and Grabovski, M. (2010). Application of artificial neural network to boiler and turbine control, *Rynek Energii*, Vol. 16, No 1. ISSN 1425-5960.

Montague, G. and Morris, J. (1994). Neural network contributions in biotechnology, *Trends in biotechnology*, Vol. 12, No 8. pp. 312-324, ISSN 0167-7799.

Taufer, I., Drábek, O., Seidl, P. (2008). Umělé neuronové sítě – základy teorie a aplikace (10), *CHEMagazín*, vol. XVII, issue 1, pp. 35-37. ISSN 1210-7409.

Teixeira, A., Alves, C. and Alves, P. M. (2005). Hybrid metabolic flux analysis/artificial neural network modelling of bioprocesses, In: *Proceedings of the 5th International Conference on Hybrid Intelligent Systems*, IEEE Computer Society, Los Alamitos. ISBN 0-7695-2457-5.