УДК 332.025.12

PIECEWISE-LINEAR NEURAL MODELS FOR PROCESS CONTROL

DOLEŽEL P., Ph.D. student; TAUFER I., prof., D.Sc; MAREŠ J., Ph.D.
(*University of Pardubice, Faculty of Electrical Engineering and Informatics, Department of Process Control, Pardubice, Czech Republic*)

INTRODUCTION

Artificial Neural Network (ANN) is a popular methodology nowadays with lots of practical and industrial applications. 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.

1 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 [1]. 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.

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.

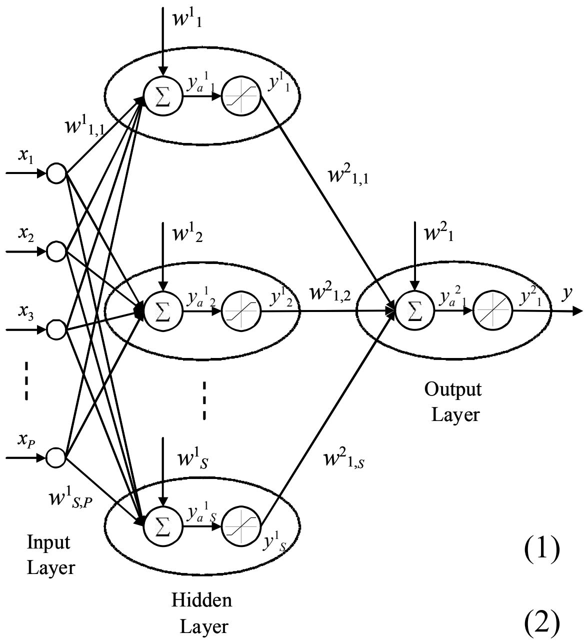

Fig. 1. Three-layered ANN

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

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

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

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

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.

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

$$y = y_a{}^2{}_1 \tag{6}$$

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.) [2].

2 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.

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

In equation above, ψ 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).

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

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 [2].

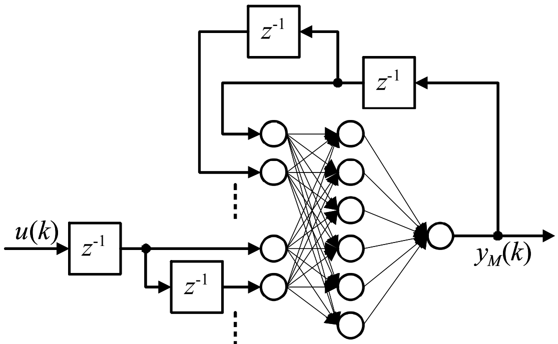

Fig. 2. Neural model

3 PIECEWISE-LINEAR MODEL

As mentioned in section 1, 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.

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

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. 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 **z** of S elements. This vector indicates saturation states of hidden neurons – see (eq. 10).

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

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

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

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

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

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

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

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

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 described above, 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.

4 EXAMPLE

Exemplary nonlinear controlled system is defined by difference equation (12).

$$y_S(k) = \frac{1.5 \cdot y_S(k-1) - 0.8 \cdot y_S(k-2) + 0.1 \cdot u(k-1) + 0.05 \cdot u(k-2)}{1 - 0.1 \cdot y_S(k-1) + 0.2 \cdot [y_S(k-1)]^2} + 0.6 \cdot \sqrt{u(k-1)} \qquad (12)$$

Firstly, system is controlled with PI controller tuned by trial and error. Control response for defined reference w_S is shown in Fig. 3. Then, piecewise-linear neural model and Pole Assignment technique [3] are used for control (Fig. 4). Compared each other, there comes clear improvement with piecewise-linear neural model.

5 CONCLUSIONS

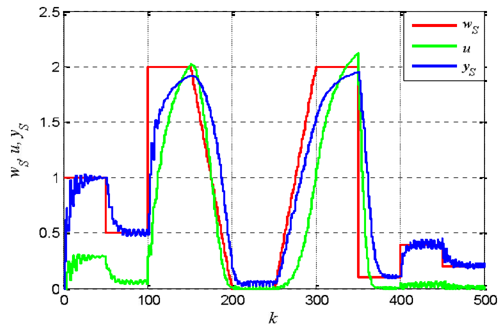

Fig. 3. Control response with PI controller

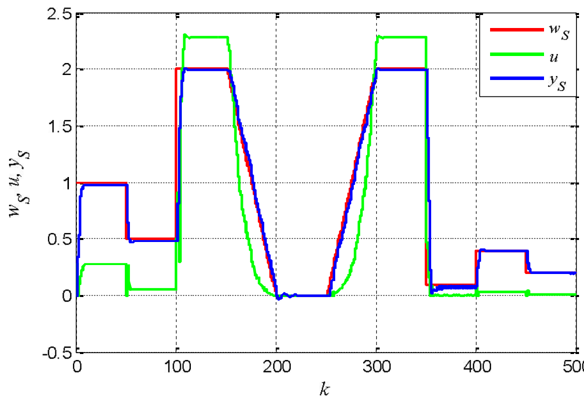

Fig. 4. Control Response with Piecewise-Linear Neural Model

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 PI controller proves great improvement. *The work has been supported by the funds No. MSM 6046137306 and No. MSM 0021627505 of Ministry of Education of the Czech Republic and No. MEB*

0810003 of Ministry of Education of the Czech Republic and Ministry of Education of the Slovak Republic.

Literatura

1.HECHT-NIELSEN, R. Kolmogorov's mapping neural network existence theorem. In: *Proc 1987 IEEE International Conference on Neural Networks.* Vol. 3, 1987, pp. 11-13. IEEE Press.
2.HAYKIN, S. *Neural Networks: A Comprehensive Foundation. Prentice Hall.* New Jersey, 1994. ISBN 0023527617
3.HUNT, K. J., Ed. *Polynomial Methods in Optimal Control and Filtering*. Peter Peregrinus Ltd. Stevenage, 1993. ISBN 0-86341-295-5.

ОЦЕНИВАНИЕ ПАРАМЕТРОВ ДИНАМИЧЕСКОЙ МОДЕЛИ ПРИ ВОЗНИКНОВЕНИИ ВОЗМУЩЕНИЙ

Елисеев А.В., к.т.н., доцент; Жуковский А.Г., к.т.н., доцент
(Ростовский военный институт ракетных войск,
Донской государственнiй технический университет г. Ростов-на-Дону, Россия)

В настоящем докладе рассматриваются основные теоретические посылки решения задачи дискретной линейной фильтрации [1] применительно к измерениям, содержащим возмущения кусочно-степенного характера, имеющие конечное число разрывов первого рода на всем интервале наблюдения [2].

Пусть вектор состояния $X(j)=X(t_{j})=\left[x(j),\ s=\overline{1,q}\right]^{T}$ объекта наблюдения на интервале $\left[t_0,\ T\right]$ описывается разностным уравнением

$$X(j+1)=\Phi(j+1,j)X(j)+\Gamma(j+1,j)N_{x}(j),\ \ j=0,1,2,..., \tag{1}$$

а наблюдаемая случайная последовательность представлена уравнением

$$Y(j)=B(j)X(j)+H(j)+N_{y}(j),\ \ j=0,1,2,..., \tag{2}$$

где $\Phi(j+1,j)=\left[\varphi\ (j),\ l,s=\overline{1,q}\right]$, $\Gamma(j+1,j)=\left[\gamma\ (j),\ s=\overline{1,q},\ k=\overline{1,m}\right]$,

$B(j)=\left[b\ (j),\ k=\overline{1,p},\ s=\overline{1,q}\right]$, $Y(j)=\left[y\ (j),\ s=\overline{1,p}\right]^{T}$ - $N(j)=\left[n\ (j),\ s=\overline{1,m}\right]^{T}$,

$N(j)=\left[n\ (j),\ s=\overline{1,p}\right]^{T}$.

Известно, что

$$M\{N(j)\}=0,\ M\{N(j)\}=0,\ M\{N(j)\,N^{T}(k)\}=V(j)\delta(j-k),$$

$$M\{N_{y}(j)\,N_{y}^{T}(k)\}=W_{y}(j)\delta(j-k),\ M\{N_{x}(j)\,N_{y}^{T}(k)\}=0, \tag{3}$$

где $V(j)=\mathrm{diag}\left[v\ (j),\ s=\overline{1,m}\right]$, $W(j)=\mathrm{diag}\left[w\ (j),\ s=\overline{1,p}\right]$,

$H(j)=\left[h\ (j),\ s=\overline{1,p}\right]^{T}$. Возмущение $H(j)$ относится к классу кусочно-степенных помех, т.е. на отрезке $\left[t_0,\ T\right]$ имеет конечное число точек разрыва первого рода и на интервалах непрерывности $\left(t_{s\,i-1},\ t_{s\,i}\right]$ описывается степенными полиномами

$$h_{si}(j)=\sum_{l=0}^{M_{si}} {}_{s,il}\left(\ -^{*}\ \right)^{l},\ \ M_{si}\in\{0,1,2,...\},\ t_{j}\in\left(\overset{*}{t}\ ,\overset{*}{t}\ \right]\subset\left[t,\ T\right], \tag{4}$$