NONLINEAR IDENTIFICATION ON BASED RBF NEURAL NETWORK

BURLAK, V. & PIVONKA, P.

Abstract: *This article is focused on the off-line identification of nonlinear dynamic systems. Hammerstein model was used for this identification. RBF (Radial Basis Function) neural network is used here to approximate the input nonlinear static function. This network is implemented as a piecewise linear function. The Simulation of identification of the mathematical model is located at the end.*

Key words: *nonlinear identification, RBF neural network, Hammerstein model, ARX model, off-line identification*

Authors´ data: Dipl.-Ing. **Burlak**, V[ladimir]; Univ.Prof. Dipl.-Ing. **Pivonka**, P[etr], Brno University of Technology, Kolejni 2906/4, 612 00, Brno, Czech Republic, xburla00@stud.feec.vutbr.cz, pivonka@feec.vutbr.cz

This Publication has to be referred as: Burlak, V[ladimir] & Pivonka, P[etr] (2011). Nonlinear identification on based RBF neural network, Chapter 44 in DAAAM International Scientific Book 2011, pp. 547-554, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-901509-84-1, ISSN 1726-9687, Vienna, Austria
DOI: 10.2507/daaam.scibook.2011.44

1. Introduction

The real systems are often nonlinear. Its static transfer characteristic is not possible to replace static a linear function. A nonlinear model must be used for their description. This model approximates it better than a linear replacement. The quality of the model is needed for designing optimal control LQ (Linear Quadratic) or predictive MPC (Model Predictive Control). A very popular approach is the distribution of nonlinear dynamic systems into two parts. The first part represents the nonlinear static function. Its output is input to the second part. The second part is a model of a linear dynamic system. This model can be found in the literature under the name of Hammerstein model (Zhu, 2001).

The process of identification is looking for a static nonlinear function and parameters of a linear dynamic model. Piecewise linear functions, polynomial or artificial neural network can be used for nonlinear static approximation. Finding the parameters of a dynamic linear model and nonlinear static function cannot be divided, because output nonlinear function is not measurable.

This article deals with the approximation of nonlinear functions piecewise linear function. This approach was realized as the RBF neural network with the extended structure.

RBF neural networks can be found in (Cichocki & Unbehanen, 1993). Used RBF neural network extensions of linear neurons are described here. Publications (Basu et al., 2007) and (Gan et al., 2008) are devoted to the issue of search centers. The issue of LQ control is described for example in (Lin, 2007). The pseudo-state representation can be found in (Bohm, 2011).

2. Model of the dynamic system

This article is focused on the dynamic systems, which can be divided on two parts. The first part is a static nonlinear amplification. Its output is input to the second part. This second part is a linear dynamic model. Further, we limit ourselves on the SISO systems (Single Input Single Output) systems.

ARX (Auto Regressive eXogenous) model was used here to describe the linear dynamic system. This model can be described by the equation (1).

$$\begin{aligned} y(k) + a_1 y(k-1) + \cdots + a_{n_a} y(k-n_a) = \\ = b_1 u(k-1) + \cdots + b_{n_b} u(k-n_b) + e(k) \end{aligned} \qquad (1)$$

The variable $u(k)$ is the value of the input signal in time k, $y(k)$ is the value of the output signal in time k and $e(k)$ is a non-measurable disturbance. Now, we can create vectors φ_l and θ. The vector φ_l is in the form (2) and θ is in the form (3).

$$\varphi_l(k) = [u(k-1) \quad \ldots \quad u(k-n_b) \quad y(k-1) \quad \ldots \quad y(k-n_a)]^T \qquad (2)$$

$$\theta = [b_1 \quad \ldots \quad b_{n_b} \quad a_1 \quad \ldots \quad a_{n_a}]^T \qquad (3)$$

These vectors can be used to rewrite the equation (1) to the vector form (4) and for building pseudo-state equations of a dynamic system. The pseudo-state description of the dynamic system is described below.

$$y(k) = \theta^T \varphi_l(k) + e(k) \tag{4}$$

Now we consider a static nonlinearity at the input of the linear system. The equation (1) is rewritten to the form (5)

$$y(k) + a_1 y(k-1) + \cdots + a_{n_a} y(k-n_a) = b_1 g\big(u(k-1)\big) + \cdots + b_{n_b} g\big(u(k-n_b)\big) + e(k) \tag{5}$$

where $g\big(u(k)\big)$ is a nonlinear static gain. When we compare the equations (1) and (5) we get the vector $\varphi(k)$ in the form (6)

$$\varphi(k) = \big[g\big(u(k-1)\big) \quad \dots \quad g\big(u(k-n_b)\big) \quad y(k-1) \quad \dots \quad y(k-n_a)\big]^T = [v(k-1) \quad \dots \quad v(k-n_b) \quad y(k-1) \quad \dots \quad y(k-n_a)]^T \tag{6}$$

where $v(k)$ is immeasurable output of the input nonlinear function. The vector of parameters θ, the nonlinear function $g\big(u(k)\big)$ and the inverse function to the function $v(k) = g\big(u(k)\big)$ must be found.

3. RBF neural network

RBF neural network can be used as a universal approach to approximate the normal functions. This network is often composed of three layers. The first layer realizes difference between the input values and its weights. These weights represent the centers of basis functions. These functions are contained in the nonlinear neurons from the second layer. These functions have usually a maximum value at its center and their value decreases with increasing distance from it. The output layer is realized the neurons with a linear transfer function. The output value is given by a linear combination of output values of neurons of the second layer. This topology can be modified by extending the second layer of parallel linear neurons. Outputs of the second layer are given by products of the linear and nonlinear neurons. The output of the network is given by (7). This modification can be found for example in (Cichocki & Unbehanen, 1993).

$$y_{NN}(k) = \sum_{i=1}^{n} f_i\big(\|x_{NN}(k) - x_i\|_{L_i}^2\big)(x_{NN}(k) - x_i)^T \theta_i \tag{7}$$

where $x_{NN}(k)$ is the input vector of the model in the step k, x_i is the center of the ith function f_i, L_i is a symmetric positive definite matrix and θ_i is the vector of the weights of the linear neuron. RBF network was used for the realization of the nonlinear input function.

Therefore, the vector of the input weights $x_{NN}(k)$ is equal to the input variable and $y_{NN}(k) = v(k)$ and h_i is a positive coefficient. The equation of the radial basis function (8) was used here

$$f_i\left(\|u(k) - u_i\|_{L_i}^2\right) = \begin{cases} 1 & \|u(k) - u_i\|_2^2 < \left\|u - u_j\right\|_2^2 \; j = 1,2,\dots n \quad i \neq j \\ 0 & otherwise \end{cases} \tag{8}$$

where $u(k)$ is the input of the neural network and u_i is the center of ith radial basis function. Now, we can determine the value of the output nonlinear static gain from the equations (7) and (8). This value is given by (9)

$$v(k) = \sum_{i=1}^{n} f_i(\|u(k) - u_i\|_2^2)(k_i u(k) + q_i) \tag{9}$$

The approximation of the static gain is realized by the piecewise linear functions. It is evident from relations (8) and (9). This option is useful for very easy finding the inverse of $g\big(u(k)\big)$. Relationship (9) can be used to find the inverse function centers v_i where $u(k) = u_i$.

4. Optimization of the model

The problem of finding the parameters of the model is consisted of two parts. The first part is looking for the centers of the radial functions. Finding the appropriate centers of the radial functions is essential for learning RBF neural network. It is advantageous to set the positions of centers of radial functions before looking for other parameters. K-means clustering, hierarchical clustering or uniform distributed centers can be used to determine the centers of the radial functions. These problems are described in (Basu et al., 2007). Determine the positions of the centers before looking for other parameters is critical for this model. Now, we can start looking for other parameters. These parameters are components of the vector β_H. This vector is in the form (10).

$$\beta_H = [k_1 \quad \dots \quad k_n \quad q_1 \quad \dots \quad q_n \quad b_1 \quad \dots \quad b_{n_b} \quad a_1 \quad \dots \quad a_{n_a}]^T \tag{10}$$

A number of unknown parameters can be reduced. The parameters q_i for $i = 2, \dots n$ can be determined by the solving of the systems of the linear equations (11) with the parameter q_1.

$$\begin{gathered} q_2 = k_1\varepsilon_1 - k_2\varepsilon_1 + q_1 \\ \vdots \\ q_n = k_1\varepsilon_1 + k_2(\varepsilon_2 - \varepsilon_1) + \dots + k_{n-1}(\varepsilon_{n-1} - \varepsilon_{n-2}) - k_n\varepsilon_{n-1} + q_1 \end{gathered} \tag{11}$$

where a constant ε_n is given by equation (12). This value represents the distance between two neighboring centers.

$$\varepsilon_n = \frac{u_n + u_{n-1}}{2} \tag{12}$$

Vector of parameters can be rewritten β_H (10) can be rewritten to the form (13).

$$\beta = [k_1 \quad \dots \quad k_n \quad q_1 \quad b_1 \quad \dots \quad b_{n_b} \quad a_1 \quad \dots \quad a_{n_a}]^T \tag{13}$$

A criterial quadratic function (14) should be minimized to find the parameters β. The vector Y contains the measured output values. This vector is in the form (17). The vector $\hat{Y}$ (16) contains the output values of model. These values are given by (15).

$$I = \frac{1}{2} E^T E = \frac{1}{2} \left(Y - \hat{Y}\right)^T \left(Y - \hat{Y}\right) \tag{14}$$

$$\hat{y}(k) = k_{i_1} b_1 u(k-1) + \cdots k_{i_{n_b}} b_{n_b} u(k-1) + a_1 y(k-1) + \cdots + a_{n_a} y(k-1) (15)$$

$$\hat{Y} = [\hat{y}(k) \quad \dots \quad \hat{y}(k-m)]^T \tag{16}$$

$$Y = [y(k) \quad \dots \quad y(k-m)]^T \tag{17}$$

The gradient method Levenberg-Marquardt was used to minimize the quadratic criterial function (14). The iterative calculation of the vectors of the parameters is given by (18)

$$\beta_{l+1} = \beta_l - (J_l^T J_l + \alpha I_I)^{-1} J_l^T E_l \tag{18}$$

where α is a convergence coefficient, I_I is an identity matrix, E_l is the vector of deviations of the model output in *l*-th iteration and Jacobian J_l is given by the equation (19)

$$J_l = \begin{bmatrix} \left. \left(\frac{\partial E}{\partial \beta_1}\right)^T \right|_{\beta = \beta_l} \\ \vdots \\ \left. \left(\frac{\partial E}{\partial \beta_{n_a + n_b + 2n}}\right)^T \right|_{\beta = \beta_l} \end{bmatrix} \tag{19}$$

The values q_i for $i = 2, \dots n$ from the vector (10) must be determined from the system of the linear equations (11). More information about this gradient method can be found for example in (Nocedal & Wright, 1999).

5. Pseudo-state representation

LQ controller is the state controller. Its output value is given by a linear combination of state variables of linear dynamic system (Lin, 2007). The design of LQ control can be done for the linear dynamic part. The controller output is recalculated through the inverse of the nonlinear input function $v(k) = g\big(u(k)\big)$. This value represents the input signal of the controlled system.

Pseudo-state representation can be used to describe a dynamic model ARX in the state space. For using model for control design, the linear dynamic part was described in this representation. This representation uses the delayed input and output values as state variables. These state variables are measurable. It is the great advantage. The pseudo state equations of ARX model are in the form (20)

$$\varphi_l(k+1) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ I_1 & 0 & 0 & 0 \\ & \theta^T & & \\ 0 & 0 & I_2 & 0 \end{bmatrix} \varphi_l(k) + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} v(k)$$

$$\hat{y}(k) = \theta^T \varphi_l(k) \tag{20}$$

where I_1 is the identity matrix with rank $n_b - 1$ and I_2 is the identity matrix with rank $n_a - 1$. These equations can be rewritten in the form (21).

$$\varphi_l(k+1) = A\varphi_l(k) + Bv(k)$$

$$\hat{y}(k) = C\varphi_l(k) \tag{21}$$

More information about the pseudo-state representation can be found for example in (Bohm, 2011).

6. Simulation results

The dynamic system with the transfer function in the form (22) was chosen for identification. The nonlinear static function was added to its input. This nonlinearity was described by the function in the form (23).

$$F(s) = \frac{10}{(10s+1)(s+1)^2} \tag{22}$$

$$v(k) = 5\left(1 - \frac{1}{2}e^{\frac{-u(k)}{5}}\right) - 2.5 \tag{23}$$

A sample period was chosen $T_{vz} = 0.5s$. The transfer function (31) was discretized with this sampling period. The function (23) is a continuous monotonic function, which passes through zero. The centers spaced evenly in the interval $u\epsilon[-5;5]$. This interval corresponds to the selected saturation of the input model. The number of the centers was equal to six. The system was excited by white noise.

The original static nonlinear function and found approximation by the piecewise linear function is shown in Fig. 1. The interval of the input values should be distributed on the smaller parts for the better approximation. However, many small parts not suitable for input signals with noise. The quality of approximation of the nonlinear function is influenced by the accuracy of parameters q_i $i = 2,3, \dots$ and by the input signal as well.

Fig. 2 shows the response of linear dynamic parts. From these curves it is evident that finding the parameters of this section leads to the very accurate results.

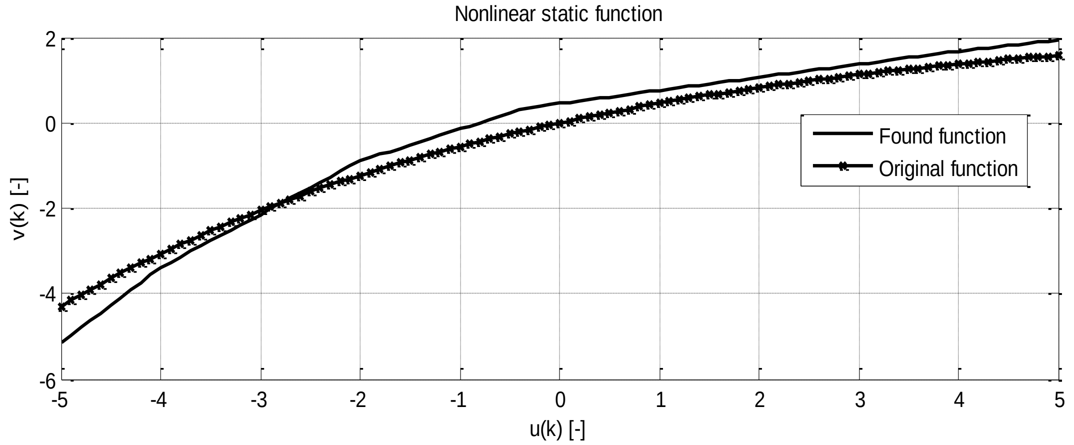

Fig. 1. Nonlinear functions

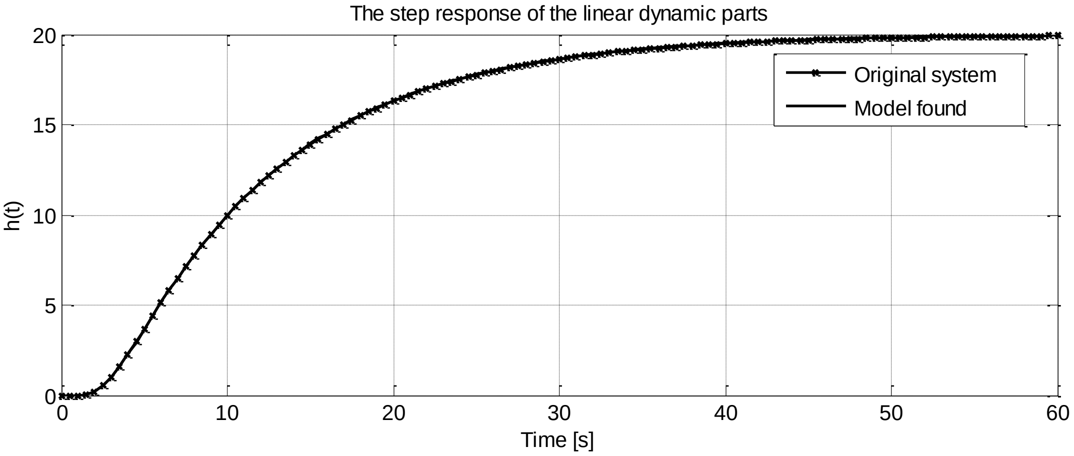

Fig. 2 Step response of linear dynamic parts

7. Conclusion

This article is focused on the nonlinear dynamic systems that can be divided into the linear dynamic part and the nonlinear static part. The output of the nonlinear part is the input of the linear dynamic part. RBF neural network was used to approximate the static nonlinear function. This network contains the neurons with the nonlinear activation functions in the form (8). The approximation of the piecewise linear functions was achieved by this option. The problem of this topology is to find the weights of the input layer neural network. The input weights are the centers of the nonlinear basis functions of the second layer neurons. These centers should be known (at least approximately) before the optimization of other parameters for effective learning. This effect complicates using of the model for on-line identification. Therefore, an algorithm for off-line identification was used here. Ambiguous gain between the two parts is a problem of this approach. The solution to this problem and finding the centers of the basis functions will be subject to further work. It is anticipated the using of this model for the online identification. The selected neural network achieves better results than the RBF network with Gaussian basis functions. The numerical method Levenberg-Marquardt was used to find the parameters here.

8. Acknowledgment

This work was supported as a part of the solution of Czech Science Foundation GAČR project No. 102/09/1680 Control Algorithm Design by Means of Evolutionary Approach and by the Czech Ministry of Education in the frame of MSM 0021630529 Intelligent Systems in Automation.

9. References

Basu, S; Davidson, I & Wagstaff, K (2008). *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, Chapman & Hall/CRC, ISBN: 1584889969

Bohm, J (2011). Adaptive LQ controllers, *Available from:* http://crr.vutbr.cz/system/files/brozura_07_1104.pdf *Accessed:* 2011-06-15

Cichocki C; Unbehanen, R. (1993). *Neural network for optimalization and signal processing,* John Wiley & sons, ISBN 047193010, Stuttgart

Dreyfus G. (2005). *Neural networks: methodology and applications***,** Springer. Science, ISBN 3-540-22980-9, Germany

Gan; G., Ma; C. & Wu J. (2007). *Data Clustering: Theory, Algorithms, and Applications,* SIAM, Society for Industrial and Applied Mathematics ISBN: 0898716233

Lin, F. (2007). *Robust control*, John Wiley & Sons, ISBN: 978-0-470-03191-9, England

Nocedal, J.; Wright, S. (1999). *Numerical Optimization*, Springer. Series in Operations Research, ISBN 0-387-98793-2, New York

Zhu, Y. (2001). *Multivariable system identification for process control,* Elsevier Science, ISBN 0-08-043985-3, England