ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2011. No. 7(113)
ELEKTRONIKA IR ELEKTROTECHNIKA

T 190 ELECTRICAL ENGINEERING
ELEKTROS INŽINERIJA

Application of RBF Network in Rotor Time Constant Adaptation

P. Brandstetter, P. Chlebis, P. Palacky, O. Skuta
Department of Electronics, VSB – Technical University of Ostrava,
17. listopadu 15, 708 00 Ostrava, Czech Republic, phone:+420 597 324 477, e-mail: pavel.brandstetter@vsb.cz

crossref http://dx.doi.org/10.5755/j01.eee.113.7.606

Introduction

Artificial neural networks (ANN) are mainly used in these types of application where the realization of another methods would be very difficult, expensive or even unrealizable. In these applications there is possible to take the advantage of the main features of neural networks, namely: approximation ability of different nonlinear functions, possibility to set their parameters in virtue of the experimental or learning data set, the quickness of information processing and their robustness. There is no necessary mathematical or structure description, there is possible to solve the problem just like the black box task with their inputs and outputs [1–8].

The Radial Basis Functions (RBF) emerged as a variant of artificial neural network (ANN) in late 80`s by Broomhead and Love and their work opened another ANN frontier. RBF network is a type of ANN for applications to solve problems of supervised learning regression, classification and time series prediction. The radial basis functions have been applied in the area of neural networks where they may be used as a replacement for the sigmoid hidden layer transfer function in multilayer perceptrons. Radial basis functions are powerful techniques which are built into a distance criterion with a respect to the centre. Such networks have 3 layers, the input layer, the hidden layer with the RBF non-linearity and the linear output layer. RBF networks have the advantage of non suffering from local minima in the same way as multilayer perceptrons. The most popular choice for the non-linearity is the Gaussian. The output layer is in regression problems a linear combination of hidden layer values representing mean predicted output [5, 6].

In most cases, it presents higher training speed when compared with ANN based on back-propagation training methods, easier optimization of performance since the only parameter that can be used to modify its structure is the number of neurons in the hidden layer etc…

Rotor time constant adaptation methods are used in the modern control of induction drive. The value of rotor resistor changes in dependence on drive load. To improve the motor power its necessary the identification of these parameters and adjusts them [1–3].

Intention of this paper is to introduce the way how new types of artificial neural networks can be chosen in the control of electrical drives. The procedure is demonstrated through the use of rotor time constant adaptation method in the vector control of an induction motor.

Vector control of the induction motor

The main problem of the vector control in the field coordinates of the induction motor is the separation of torque and flux control circuits not to be mutually influenced. The torque of the induction motor and consequently the active power are controlled by the torque control circuit while the rotor flux and consequently reactive power are controlled by the rotor flux control circuit.

The whole control operates on the principle of stator current space vector decomposition into two perpendicular elements i_{Sx} and i_{Sy} which can be analyzed in the field coordinate system [x, y] with rotor flux space vector orientation to the x axis (Fig. 1) [2, 4].

Independent quantities - torque and magnetization can be analyzed by this separation. By maintaining the amplitude of the rotor flux ($\psi_R=K_\psi i_m$) at a fixed value there is a linear relationship between torque t and the torque component i_{Sy} ($t=K_t\,\psi_R\, i_{Sy}$).

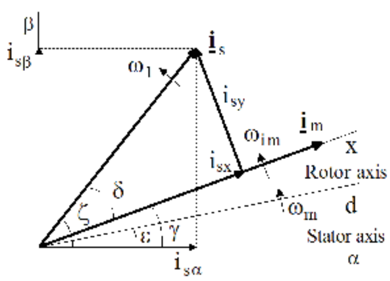

Fig. 1. Structure of the current model

Rotor time constant adaptation

The induction motor with vector control has a very good dynamic behavior and as a consequence is well suited for high performance applications. But, the vector control is very sensitive to variations in the rotor time constant. The decoupling between the flux and torque is lost in an indirect rotor field oriented control if there is a mismatch between the controllers set up rotor time constant and the actual time constant of the motor. Adaptation of this rotor time constant is thus required, and it is necessary to estimate this parameter in order to maintain it equal to its rated value programmed in the decoupling controller.

Model reference adaptive system method

The block structure of the model reference adaptive system (MRAS) with the adaptation method of rotor time constant is shown in Fig. 2. The method is based on the comparison of two estimators, where one of them includes rotor time constant, which is called the adaptive model. The other one does not include rotor time constant and is the so-called reference model. The error between them is used to derive an adaptation algorithm which produces the estimated value of a rotor time constant for the adaptive model. This value can be used for adaptation of a rotor time constant in the current model, which is used in the control structure of induction motor drive.

The adaptive model is based on the application of a current model of rotor flux. We often use it for the determination of the value and position of the magnetizing current vector or rotor flux vector. The current model contains the rotor time constant which is a changing parameter. The adaptive model is described as follows

$$\hat{\overline{\Psi}}_R^S = \int\left[\left(j\omega_R - \frac{1}{\hat{T}_R}\right)\hat{\overline{\Psi}}_R^S + \frac{1}{\hat{T}_R}L_m\overline{i}_S^S\right]dt\ , \qquad (1)$$

where $T_R = L_R / R_R$ is rotor time constant, R_R is resistance of rotor winding, L_R is rotor inductance, ε is rotor position angle, $\omega_R = d\varepsilon/dt$ rotor angular speed.

The reference model is based on application of voltage model of rotor flux and is described as follows

$$\overline{\Psi}_R^S = \frac{L_R}{L_m}\left[\int\left(\overline{u}_S^S - R_S\overline{i}_S^S\right)dt - \frac{L_S L_R - L_m^2}{L_R}\overline{i}_S^S\right]. \qquad (2)$$

The quantities $\overline{q}^S$ are vectors in stator reference frame: rotor flux vector $\overline{\psi}_R^S = \psi_{R\alpha} + j\,\psi_{R\beta}$, stator voltage vector $\overline{u}_S^S = u_{S\alpha} + j\,u_{S\beta}$, stator current vector $\overline{i}_S^S = i_{S\alpha} + j\,i_{S\beta}$, R_S is resistance of stator winding, L_S and L_R are the stator and rotor inductances, L_m is the magnetizing inductance.

The adaptation algorithm it is described by the following equations:

$$\Phi(e) = e_\alpha\left(L_m i_{S\alpha} - \hat{\Psi}_{R\alpha}\right) + e_\beta\left(L_m i_{S\beta} - \hat{\Psi}_{R\beta}\right), \qquad (3)$$

$$e_\alpha = \left(\Psi_{R\alpha} - \hat{\Psi}_{R\alpha}\right), \quad e_\beta = \left(\Psi_{R\beta} - \hat{\Psi}_{R\beta}\right), \qquad (4)$$

$$\frac{1}{\hat{T}_R} = K_1\Phi(e) + K_2\int\Phi(e)\,dt\ , \qquad (5)$$

where $K_1 > 0$, $K_2 > 0$.

The adaptation mechanism consists of evaluation of adaptation signal (3) and its sequential minimalization by the help PI-controller (5). Fig. 2 shows the structure of model adaptive reference system and also the substitution of the adaptation mechanism with the artificial neural network.

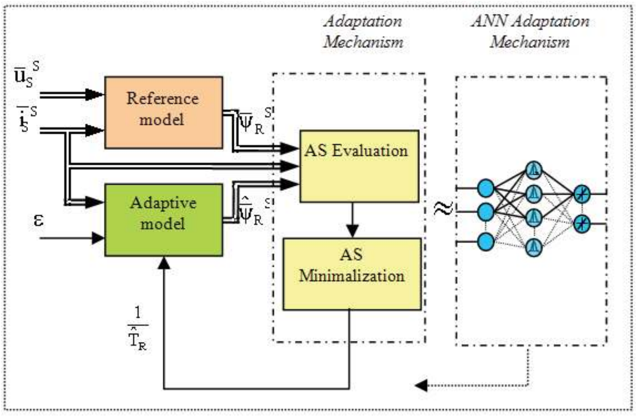

Fig. 2. Structure of model reference adaptive system

Radial basis function network

The aim of this work was to compare the features of Radial Basis network with different architectures. In this paper the effort focused in different architectures of RBF, also there was added white noise, which is very useful in the application of feed-forward neural networks.

There was realized comparative procedure. At first there was realized common RBF network with the appropriate architecture, it mean with one, two or without feedback, etc… Then there was changed the field of coverage from one RBF unit. In fact it means more sporadic or densely lay-out of the RBF units, which is expressed by lower or higher number of RBF units.

The figure 3 depicts the data acquisition of training data set for the off-line neural network training. The start of the motor was set without load and in the time 0.5 seconds with the load. The model was always adjusted according to the actual architecture of tested RBF network.

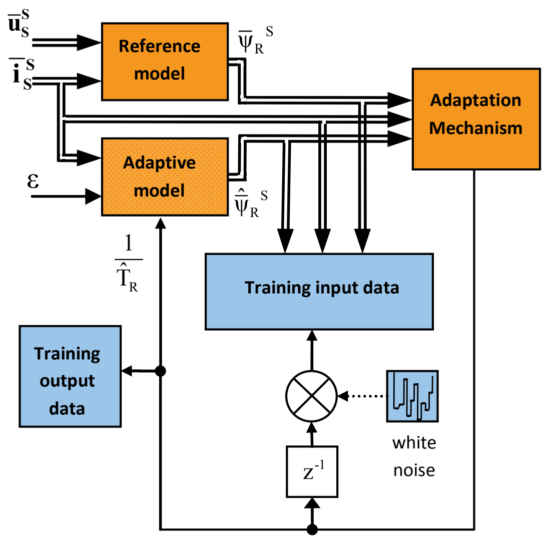

Fig. 3. Block structure of the in-out data training acquisition

Different types of training algorithms were tested and evaluated as the most fitting. Three training algorithms were used to test the main features of RBF neural networks:

- Forward subset selection;
- Ridge regression;
- Regression trees 1 & 2.

From these training algorithms there were picked like a useful for ours purpose just the Forward subset selection algorithm. This algorithm was variously modified together with changes of the RBF network (e.g. activation function, radius…). The other methods should be useful for some other problems.

RBF network with one feedback

The first type is the most used and common RBF network with one feedback without scaling and without the white noise.

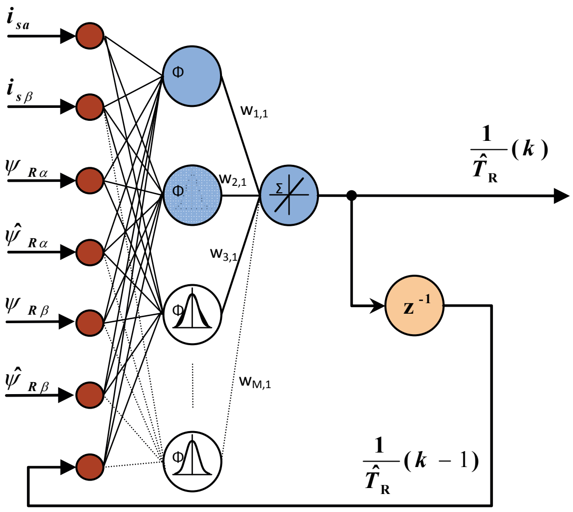

Fig. 4. Architecture of RBF neural network

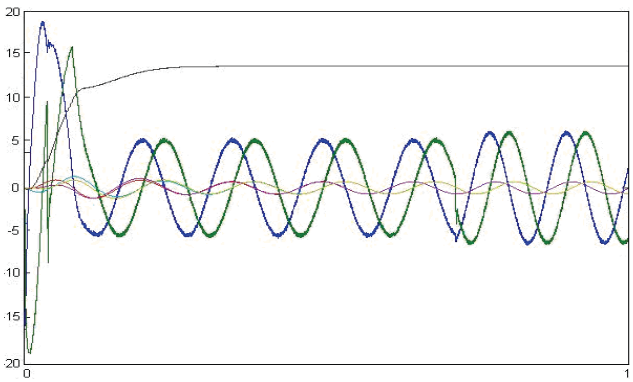

Fig. 5. Input training data set

The Fig. 4 depicts the RBF architecture with the appropriate input variables. There are always three layers: input, hidden layer with the non-linear activation function and the output linear layer.

There are the input data for the adaptation mechanism, which were also used like an input training data set for the neural network, in the Fig. 5 ($i_{S\alpha}$, $i_{S\beta}$, $\psi_{R\alpha}$, $\psi_{R\beta}$, $\hat{\psi}_{R\alpha}$, $\hat{\psi}_{R\beta}$ $1/T_{RRBF\text{-}k} = f(t)$ [~,s]).

Output or we can say the desired output time behavior is always depicted in the Fig.s by the red dotted line.

In the first RBF neural network there were used 97 RBF units. The output time behavior is perfect as we can see in the Fig. 6, the difference between the adaptation mechanism (AM) and the RBF network (RBFN) is really neglect able (Fig. 7).

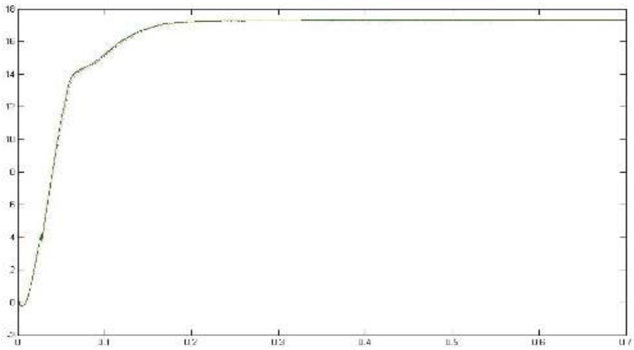

Fig. 6. Output signal $1/T_R = f(t)$ $[s^{-1},s]$ from RBFN and AM

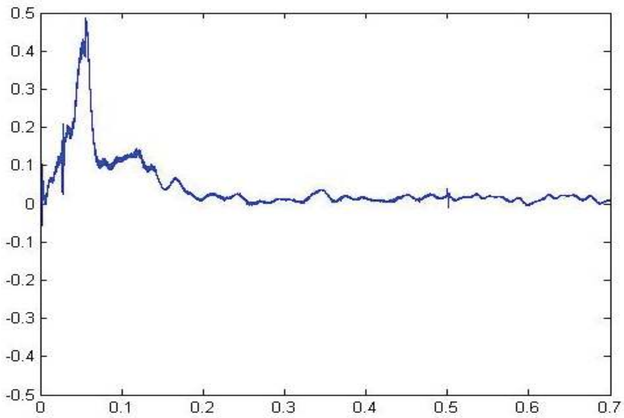

Fig. 7. Difference between RBFN and AM output signal

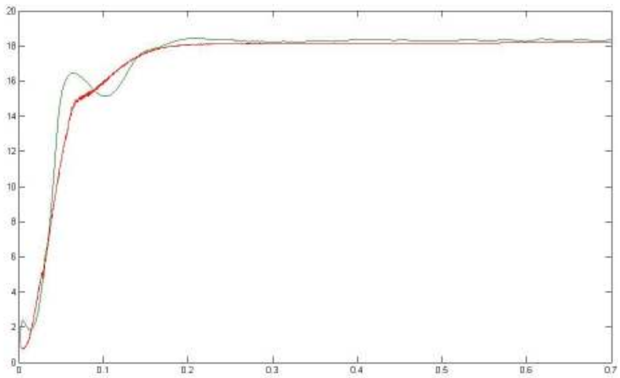

Fig. 8. Output signal $1/T_R = f(t)$ $[s^{-1},s]$ from RBFN and AM

Next network was used with thinly lay-out of 33 RBF units and the response is also quite good (see Fig.8).

Last one was used with denser lay-out of 365 RBF units and the output was almost the same like with the 96 RBF units. Then is no reason to use this kind of structure because of higher memory demand and higher computation time.

RBF without feedback connection

Next architecture of RBF network didn't include the feedback. In the Fig. 10 there is possible to see that this output behavior is not the expected one. The network contains 81 RBF units. In the next Fig. there is obvious improvement of the output curve, but the price was higher number (261) of the RBF units.

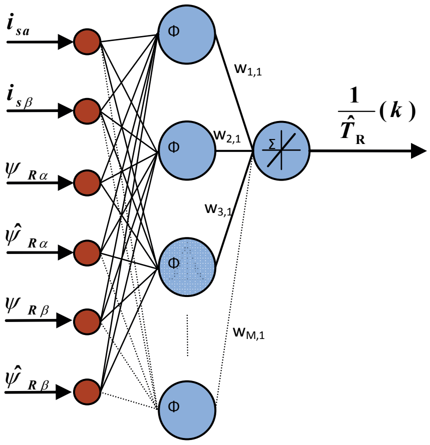

Fig. 9. Architecture of RBF neural network

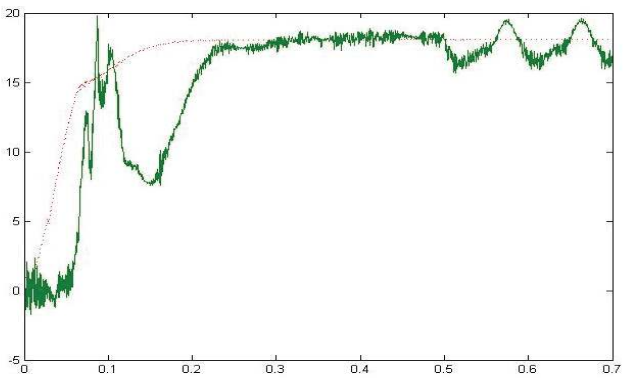

Fig. 10. Output signal $1/T_R = f(t)$ $[s^{-1}, s]$ from RBFN and AM

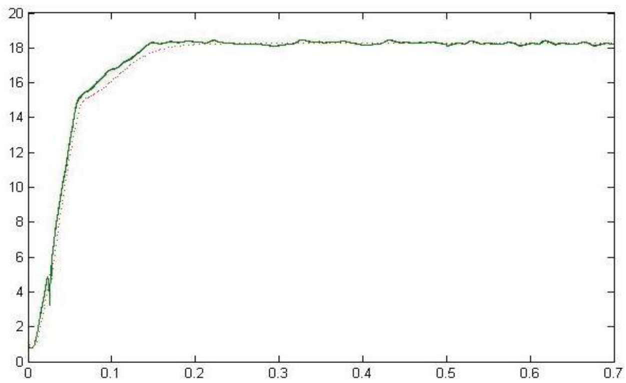

Fig. 11. Output signal $1/T_R = f(t)$ $[s^{-1}, s]$ from RBFN and AM

RBF with two feedback connection

There is described the RBF architecture with two feedbacks connections. As we can see the output time behaviour is also very good (Fig.12 & Fig. 13) like in the first case. There were used 120 RBF units and the next was used with thinly lay-out of 51 RBF units and the response is also quite good (Fig.14). Denser lay-out of RBF network disposes with 261 RBF units and this is the same problem like with network with one feedback.

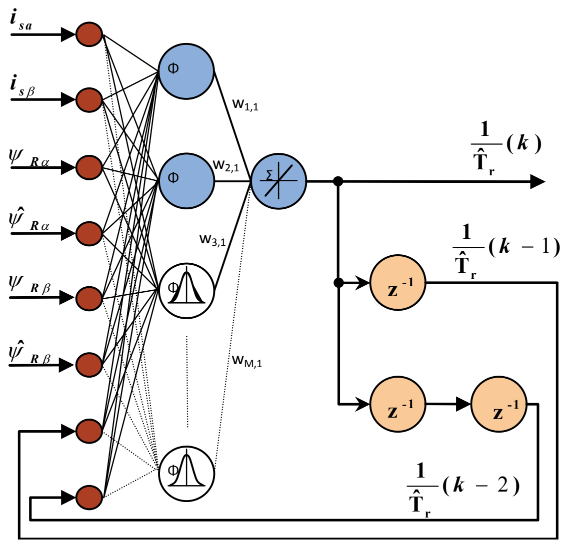

Fig. 12. Architecture of RBF neural network

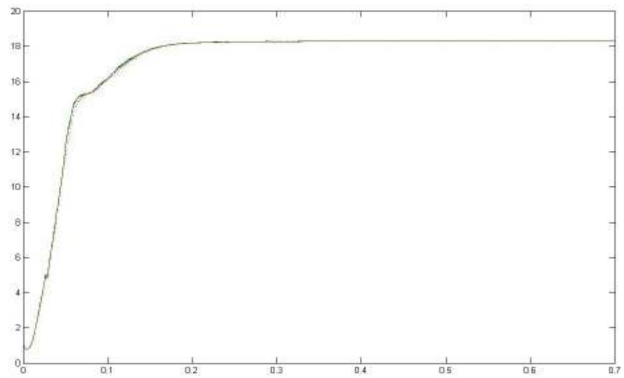

Fig. 13. Output signal $1/T_R = f(t)$ $[s^{-1}, s]$ from RBFN and AM

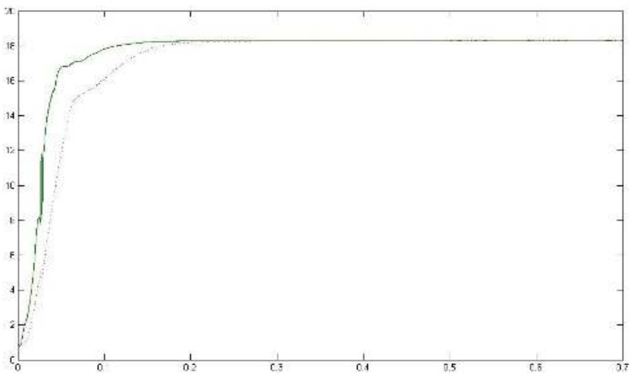

Fig. 14. Output signal $1/T_R = f(t)$ $[s^{-1}, s]$ from RBFN and AM

RBF with the scaled input variables

The next architecture comes from idea of feed-forward architecture, where the input values must be scaled because of their activation function. In the Fig. 15 there are depicted input scaled training data set for RBF neural network ($i_{S\alpha}$, $i_{S\beta}$, $\psi_{R\alpha}$, $\psi_{R\beta}$, $\hat{\psi}_{R\alpha}$, $\hat{\psi}_{R\beta}$ $1/T_{RRBF\text{-}k} = f(t)$ $[\sim, s]$).

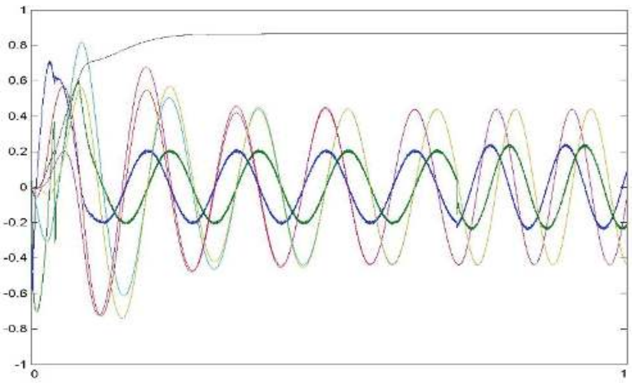

Fig. 15. Input training data set

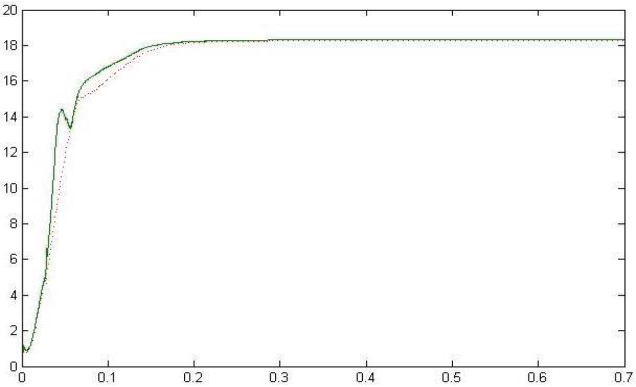

Fig. 16. Output signal $1/T_R = f(t)$ $[s^{-1},s]$ from RBFN and AM

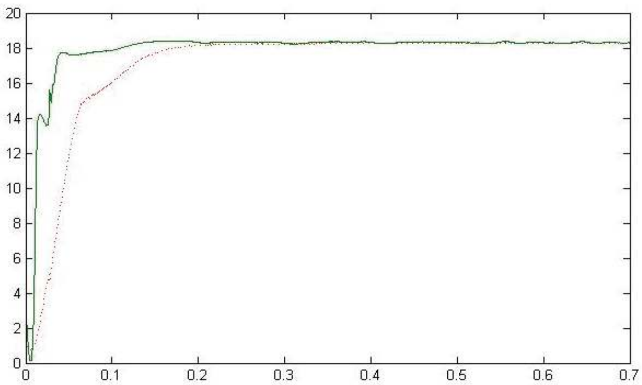

Fig. 17. Output signal $1/T_R = f(t)$ $[s^{-1},s]$ from RBFN and AM

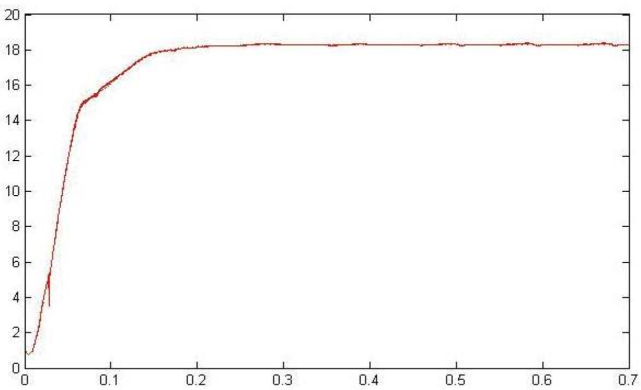

Fig. 18. Output signal $1/T_R = f(t)$ $[s^{-1},s]$ from RBFN and AM

Only the dense (436 units) RBF unit lay-out provide output curve (fig.18) like the unscaled networks. The classical one with 116 units we can also consider good enough (fig.16), but network with thin lay-out (32units) has low-quality output curve (fig.17). There were one important difference in lower values of the inner network parameters like radius, centers and weights.

RBF with the white noise

With an addition of the bounded white noise there were idea of reduce the weight and centers importance of the feedback connection. In the feed-forward neural networks it has the less neural hidden unit foundation. The input training data set is depicted in Fig. 19 ($i_{S\alpha}$, $i_{S\beta}$, $\psi_{R\alpha}$, $\psi_{R\beta}$, $\hat{\psi}_{R\alpha}$, $\hat{\psi}_{R\beta}$ $1/T_{RRBF\text{-}k} = f(t)$ $[\sim,s]$).

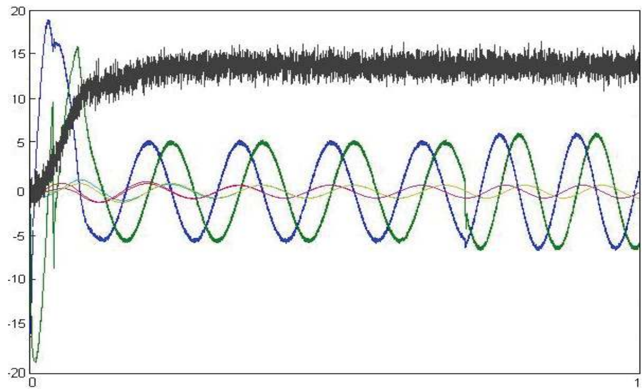

Fig. 19. Input training data set

There was used just only one type of RBF architecture with the classic lay-out of RBF units. The RBF neural network contains 215 activation units and then was useless to go on with this type. It will be discussed in the conclusion. Anyway, in the Fig. 20 there are depicted almost perfect output curves. It shows us that the difference between the "reference" adaptive model and the RBF network could be neglected.

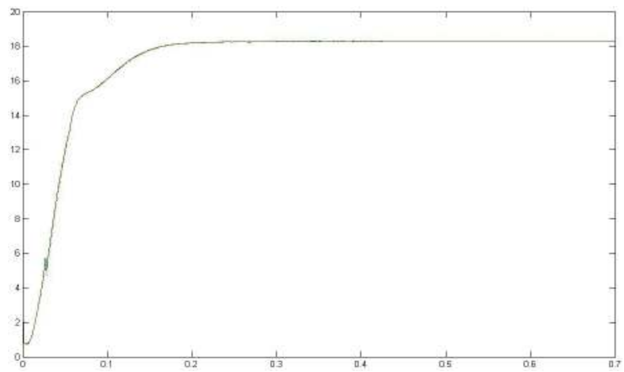

Fig. 20. Output signal $1/T_R = f(t)$ $[s^{-1},s]$ from RBFN and AM

Conclusions

The paper deals with different architectures of Radial Basis Function neural network. At the end of this paper, there must be sad, that the most common architecture of RBF network with one feedback connection presents the best output time behavior in comparison with the others. RBF without feedback connection presents quite unstable and inaccurate output. The RBF with two feedbacks has very good output curves, but if we realize higher number of RBF units and more complicated connections then the result is also against this type. The next architecture with scaled input neither had better time behavior. The only result was lower values of the hidden layer variables like radius, centers and weights.

The last type with used white noise gives us lower number of hidden units in the feed-forward neural networks, but it does not work with RBF network. That is why there were not used other types with different layout.

The result of this paper is, that there could be use other types of RBF architecture if is necessary for some reason, like scaled input variables or non-present feedback connection, but then must be considerate the mentioned disadvantages.

Some of these more interesting theoretical assumptions were verified on real laboratory model with induction motor controlled by digital signal processor with the system for the training data acquisition.

Acknowledgements

In the paper there are the results of the project 102/08/0755 which was supported by The Czech Science Foundation (GA CR).

References

1. **Aghion H., Ursaru O., Lucanu S.** Three – phase motor control using modified reference wave // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 3(99). – P. 35–38.
2. **Brandstetter P.** A. C. Control Drives – Modern Control Methods. – VSB–Technical University of Ostrava, 1999.
3. **Brandstetter P., Simonik P.** Signal Processing for Vector Control of AC Drive // Conference Proceedings, 20th International Conference Radioelektronika. – Brno, Czech Republic, 2010. – P. 93–96.
4. **Dobrucky B., Spanik P., Kabasta M.** Power electronic two–phase orhogonal system with HF input and variable output // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 1(99). – P. 9–14.
5. **Haykin S.** Neural Network a Comprehensive Foundation. – New Jersey: Prentice–Hall, 1999.
6. **Orr J. L.** Introduction to Radial Basis Function Networks. – University of Edinburg, 1996.
7. **Perdukova D., Fedor P.** Fuzzy Model Based Control of dynamic System // JEE–Journal of Electrical Engineering. – University Politechnica Romania, 2007. Vol. 7. – No. 3. – P. 5–11.
8. **Vas P.** Artificial–intelligence–based Electrical Machines and Drives, 1st ed. – Oxford: University Press, 1999.

Received 2011 11 10

P. Brandstetter, P. Chlebis, P. Palacky, O. Skuta. Application of RBF Network in Rotor Time Constant Adaptation // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 7(113). – P. 21–26.

The paper presents the results of the rotor time constant adaptation method with the application of artificial neural network. The estimation of the rotor time constant for adaptive model of MRAS is realized by the help of PI-controller and then is replaced by the Radial Basis Function network. The estimated rotor time constant is then used in the vector control of electrical drive. There were discussed the different architectures of RBF network in the field of adaptation of rotor time constant parameter. Simulations have been performed in the Matlab-Simulink. Ill. 20, bibl. 8 (in English; abstracts in English and Lithuanian).

P. Brandstetter, P. Chlebis, P. Palacky, O. Skuta. RBF tinklo įtaka rotoriaus laiko pastoviajai // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2011. – Nr. 7(113). – P. 21–26.

Pateiktas metodas įvertinantis rotoriaus laiko pastoviąją dirbtiniuose neuroniniuose tinkluose. Pateiktos kelios RBF tinklo struktūros. Atliktas modeliavimas naudojantis programų paketu „Matlab“. Il. 20, bibl. 8 (anglų kalba; santraukos anglų ir lietuvių k.).