Peter Girovský

Neurocontroller for Induction Motor

The paper is focused especially on presenting possibilities of applying artificial neural networks (ANN) at creating the system inverse models that are expected to be of use at designing control algorithm for non-linear dynamic systems. The paper deals with a quasi-inverse model which is intended to work as a controller of a non-linear dynamic system simulating an induction motor. The presented method of control takes advantage of approximating properties of multi-layer perceptron (MLP) networks. Obtained simulation results are a contribution to theoretical discussion of the problem.

Keywords: neural network, control, induction motor

I. INTRODUCTION

In reality, most of the systems are non-linear. However, many systems can be represented without any significant loss of accuracy by an equivalent linear representation. Control designs based on system linearization are a widely applied technique in the industry. Other systems are increasingly characterized by complex non-linear dynamics (e.g. high non-linearity, abrupt parameter variations, external disturbances etc.). The important non-linear diversity is the main reason why no systematic and generally applicable theory for non-linear control design has been developed yet. It is the ability of the artificial neural networks to model non-linear systems that can be the most readily exploited in the synthesis of non-linear controllers.

We are interested in different already existing methods used to develop a neural controller based on an inverse model of a system.

Part one of the paper is focused on explaining the method [1], while the following one demonstrates the use of ANN for purposes of controlling in simulation studies for a squirrel-cage induction motor. Last part of the paper demonstrates sensitivity of the speed controller to motor parameters changes.

II. THE INVERSE NEURAL MODEL

Although the system inverse model plays an important part in the theory of control, the attainment of its analytical form is pretty strenuous. Anticipating that a dynamic system can be described by the differential equation

$$y(k+1)=f[y(k), ..., y(k-n+1), u(k), ..., u(k-m+1)] \quad (1)$$

where the system output y(k+1) depends on the preceding n-output and m-input values, the system inverse model can be generally presented in the following form

$$u(k)=f^{-1}[r(k+1), y(k),..., y(k-n+1), u(k), ..., u(k-m+1)] \quad (2)$$

Here $y(k+1)$ is an unknown value, and hence can be substituted by the output quantity desired value $r(k+1)$. The simplest way to arrive at a system inverse neural model is it to train the neural network to approximate the system inverse model.

In real life, the most frequently used are two concepts of inverse neural model architecture: the ”general training” architecture (Fig.1), and the “specialized training” architecture (Fig.2), respectively.

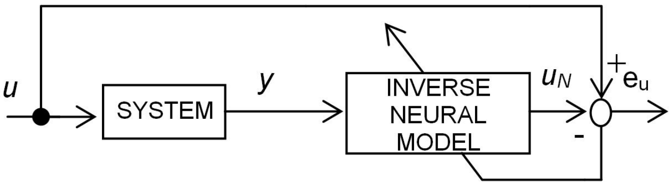

Figure 1. The ”general training” architecture.

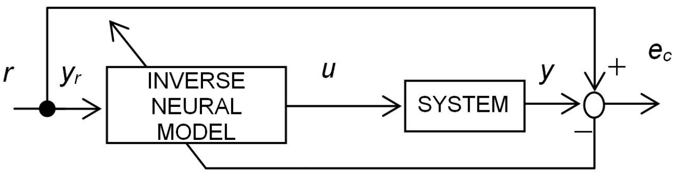

Figure 2. The ”specialized training” architecture.

If in the general training architecture, signal u is applied to the system input, signal y is obtained at the system output. The different between the incoming signal u and the neural model output u_N is the error e_u which can be utilized for network learning.

Contrary to the previous approach, it is error e_c that is in the “specialized training” architecture utilized for neural network training. Error e_c is here obtained as the difference between the desired input signal r and the signal y that represents the actual system output.

The former of the two methods brings several substantial disadvantages:

- ♣ The selection of varying output signal y values, specified for training of the network, cannot guarantee that the trained system output will fall exactly into those regions that are important for its successful utilization in controlling.
- ♣ If the controlled system happens to be multidimensional, the attained model that is represented by the inverse neural model may be incapable of imitating a real system.

The above outlined drawbacks can be circumvented by the letter method – the “specialized training” architecture-that yields some advantages when compared with the former one:

- ♣ The method (Fig.2) is intended directly for controlling, whereas the training signal is formed in dependence on the difference between the system desired and real outputs.
- ♣ In the case of multidimensional dynamic system, the real inverse model can closely simulate a real system.

Multi-layer neural networks can be utilized when creating a system inverse neural model. The use of the MLP type static neural networks presents the simplest solution, however the representation of the system dynamic remains problematic with this neural model. The application of a MLP type neural network with time delaying of the input layer signals can present the solution for introducing the process dynamics into MLP type static neural network. The solution falls among the simplest ones, and the advantage of utilizing this network type rests with the opportunity of its training by traditional backpropagation algorithm for multi-layer networks.

III. DESIGN OF THE NEUROCONTROLLER

The main requirement we have specified is maintaining the desired speed of the induction motor. Considered for the neurocontroller output were the voltage components that would present an action intervention for PWM modulation, which would eventually produce the stator voltage desired values from the mains voltage (rectified via using an uncontrolled rectifier). Since the neurocontroller output in such a structure is not directly equal to voltage fed into the motor we have abandoned the idea to establish an accurate inverse model; considered for input quantity of the quasi-inverse neural model were rather the desired and at a time also real (measured) motor speed (Fig.3).

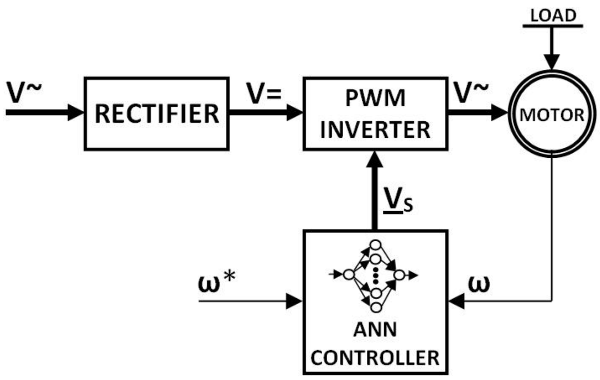

Figure 3. ANN induction motor control scheme.

The design of the neurocontroller is based rightly on known values of these speeds.

A typical technique for control synthesis purposes is based on using a description of the induction motor in rotating reference frames (x, y). The use of such rotating reference frames has the benefit of simplifying the model of the motor from the point of view of controller design. In this section design of the neural controller will be presented.

The neurocontroller consists of three multilayer perceptron (MLP) networks with backpropagation learning algorithm.

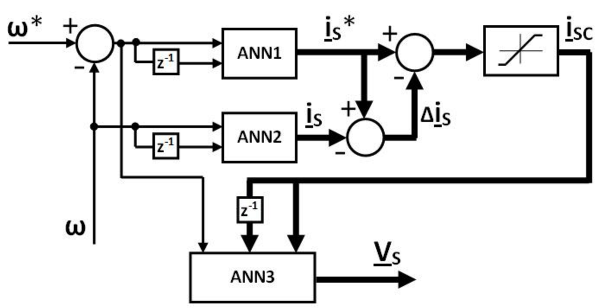

Figure 4. Concept of the ANN controller.

The first subsystem (Fig.4) of the neurocontroller serves for desired current components reconstruction and the second subsystem serves for real current components identification. The third of them serves for corresponding voltage components reconstruction for PWM converter. These voltage components present action intervention for PWM modulation that would make up the desired stator voltage values from the mains voltage (rectified using an uncontrolled rectifier). The overall control structure is shown in Fig.3.

IDENTIFICATION AND CONTROL OF STATOR CURRENTS

The first and second neural networks are identical. Their inputs are values of angular speed, expected and actual, in k-th and $(k-1)$-th step. The structure of the ANNs is shown in Fig.5.

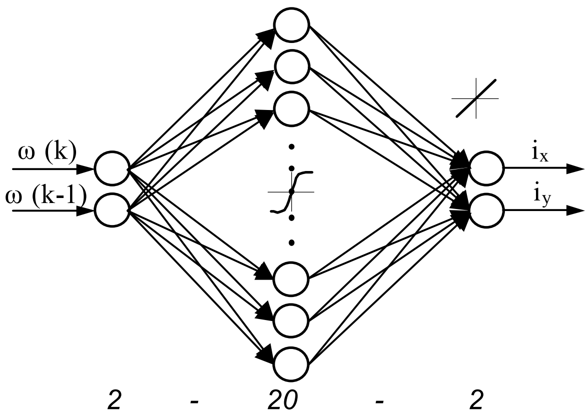

Figure 5. The structure of the ANN1 and ANN2.

$$\begin{aligned} i^*_{sx}\,(k+1) &= f[\omega^*(k),\ \omega^*(k-1),\ \mathbf{w}] \\ i^*_{sy}\,(k+1) &= f[\omega^*(k),\ \omega^*(k-1),\ \mathbf{w}] \end{aligned} \tag{3}$$

and

$$\begin{aligned} i_{sx}\,(k+1) &= f[\omega(k),\ \omega(k-1),\ \mathbf{w}] \\ i_{sy}\,(k+1) &= f[\omega(k),\ \omega(k-1),\ \mathbf{w}] \end{aligned} \tag{4}$$

Reconstructed actual stator current (3) corrects desired value of current from the first ANN (4):

$$\Delta \mathbf{i}_s\,(k+1) = \mathbf{i}^*_s\,(k+1) - \mathbf{i}_s\,(k+1) \tag{5}$$

Resulting signal of the correction:

$$\mathbf{i}_{sc}\,(k+1) = \mathbf{i}^*_s\,(k+1) - \Delta \mathbf{i}_s\,(k+1) \tag{6}$$

in k-th and $(k-1)$-th steps and the desired speed value present inputs to the third ANN, which generates appropriate voltage values for PWM converter:

$$\mathbf{u}_s\,(k+1) = g[\mathbf{i}_{sc}\,(k+1),\ \mathbf{i}_{sc}\,(k),\ \omega^*,\ \mathbf{w}] \tag{7}$$

The structure of the ANN is shown in Fig.6. MLP networks are used for all f and g approximations. The number of inputs to each of them is determined by the relation (3), (4) or (7). Twenty hidden neurons in one hidden layer of every neural subsystem employ the hyperbolic tangent functions.

All the networks are trained off-line in order to minimise the control performance. Training patterns for an ANN controller were prepare by numerical simulations of the induction motor model with help of Matlab-Simulink and Neural Network Toolbox. In simulations the nominal data of a 3kW induction motor were used. The Backpropagation training algorithm with Levenberg-Marquardt´s modification was used for the training procedures.

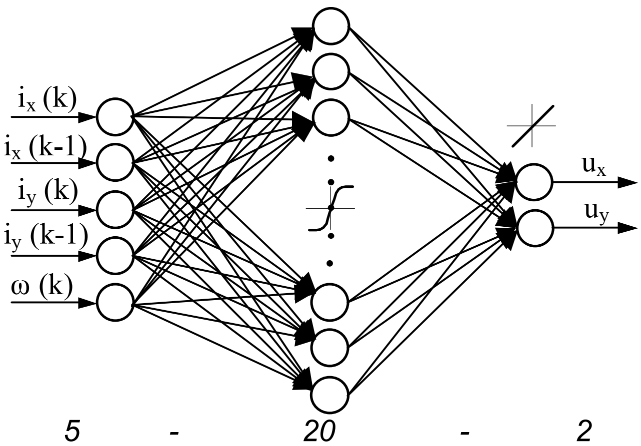

Figure 6. The structure of the ANN3.

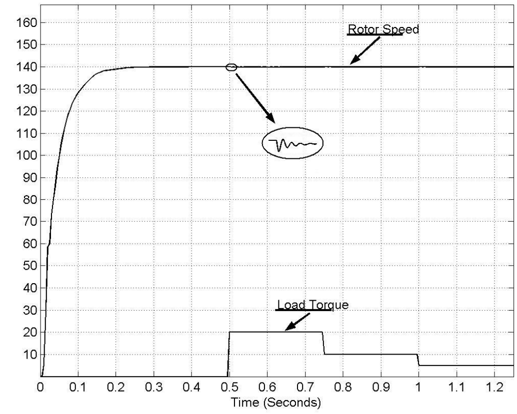

Figure 8. Motor speed at load test.

IV. SIMULATION RESULTS

Presented in this section will be the results simulated in MATLAB environment for given connection of the control diagram shown in Fig.3, where the designed neural controller was implemented. The testing of the neural controller was performed on the induction motor with the following parameters:

$U=220V/50Hz$, $I_N=6.9A$, $P=3kW$, $n_N=1420RPM$, $R_1=1.81Ohm$, $R_2=1.91Ohm$, $L_{1\sigma}=L_{2\sigma}=8.85mH$, $L_h=0.184H$, $p_p=2$, $M_N=20.17Nm$, $J=0.1kgm^2$.

In the following figures the speed control quality is presented. The neural speed controller was trained in the wide range of speed and load torque changes based on the simultation results obtained for induction motor model. Then the trained controller was tested for speed reference signal different than used in the training procedures. These testing signals together with results of simulations are presented in Fig.7 and Fig. 8. The characteristics were obtained for nominal parameters of the induction motor.

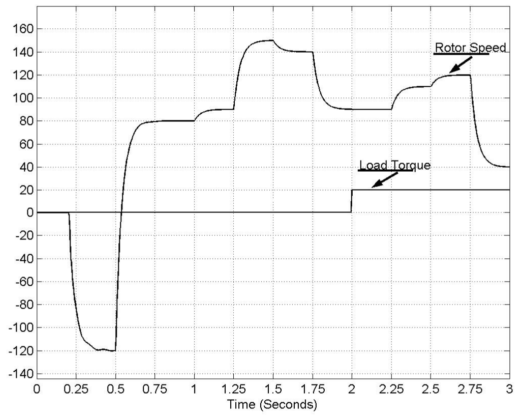

Figure 7. Motor speed at changes speed reference value and at change of load torque.

SENSITIVITY OF THE SPEED CONTROLLER TO MOTOR PARAMETER CHANGES

The trained controller was tested also for changed motor parameters different than used in the training procedures. These testing signals together with results of simulations are presented in Fig.9, Fig. 10 and Fig. 11.

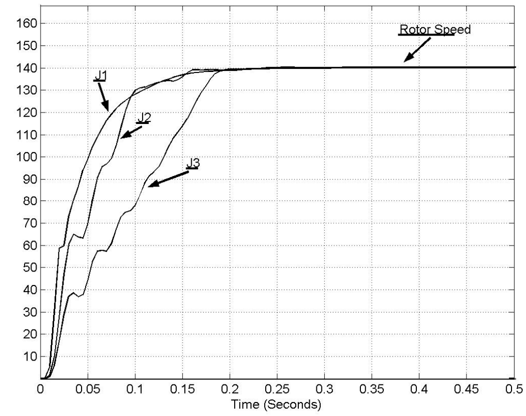

Figure 9. Motor speed for torque of inertia changes.

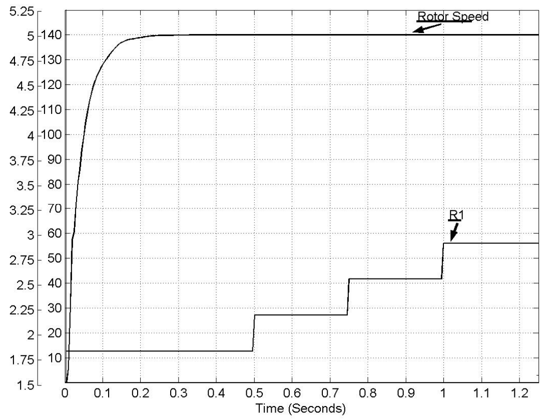

Figure 10. Motor speed for stator resistance changes.

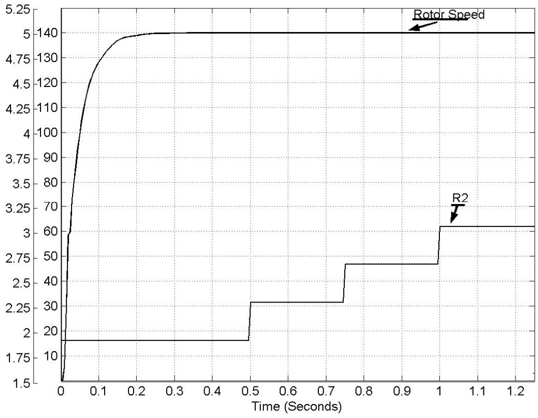

Figure 11. Motor speed for rotor resistance changes.

V. CONCLUSION

The paper deals with the design of a neural controller for non-linear dynamic system control based on a quasi-inverse model of the system. The design of neural controller is based on sensor information pertaining to angular speed of an induction motor. The control task is to reach the motor desired angular speed. The neurocontroller consists of three neural networks with backpropagation learning algorithm. First subsystem of the neurocontroller serves for desired current components reconstruction and the second subsystem serves for real current components reconstruction. The third of them serves for corresponding voltage components reconstruction for PWM converter.

MLP networks are used for all function approximations. Twenty hidden neurons in one hidden layer of every neural subsystem employ the hyperbolic tangent functions. All the networks are trained off-line in order to minimise the control performance. Training samples for an ANN controller were attained via simulation of an induction motor model in MATLAB environment. Used with these networks was learning by use of Levenberg-Marquardt algorithm. Simulations demonstrated good performance of this method.

ACKNOWLEDGMENT

This publication is the result of the Project implementation: Research centre for efficient integration of the renewable energy sources, ITMS: 26220220064 supported by the Research & Development Operational Programme funded by the ERDF.

We support research activities in Slovakia / Project is cofinanced from EU funds.

REFERENCES

[1] M. A. Brdys, G. J. Kulawski.: "Dynamic Neural Controllers for Induction Motor," IEEE Transactions on Neural Networks, 1999, Vol. 10, No.2, pp.340-355.

[2] B. Burton, R. G. Harley, G. Diana, J. L. Rodgerson: "Implementation of a Neural Network to Adaptively Identify and Control VSI-Fed Induction Motor Stator Currents," IEEE Transaction on Industry Applications, Vol.. 34. No. 3, May/June 1998.

[3] A.U. Levin, K.S Narendra,.: "Control of Nonlinear Dynamical Systems Using Neural Networks: Controllability and Stabilization." IEEE Transactions on Neural Networks, 1993, Vol. 4, pp.192-206.

[4] A.U. Levin, K.S Narendra,.: "Control of Nonlinear Dynamical Systems Using Neural Networks- Part II: Observability, Identification and Control.". IEEE Transactions on Neural Networks, 1996, Vol. 7, pp.30-42.

[5] M. T. Wishart, R. G. Harley: "Identification and Control of Induction Machines Using Artificial Neural Networks." IEEE Transaction on Industry Applications, vol. 31. No. 3. May/June 1995.

[6] J. Timko, J. Žilková, P. Girovský: "Modeling and control of electrical drives using neural networks", (in Slovak), C-Press, Košice, 2009, p. 202, ISBN 978-80-8086-124-7.

[7] J. Vittek, P. Boris, M. Štulrajter, M. Pácha: "Chattering free sliding mode control law for position control of the drive employing induction motor", Power Engineering Conference 2008, AUPEC '08, Australasian Universities, pp.1-6, 14-17 Dec., 2008.

[8] M. Kuchar, P. Brandštetter, M. Kaduch: "Sensorless induction motor drive with neural network". IEEE, Annu. Power Elec. Specialists Conf.m pp.3301-3305, 2004.

ADDRESSES OF AUTHORS

Peter Girovský, Technical University of Košice, Department of Electrical Engineering and Mechatronics, Letná 9, Košice, SK 04200, Slovak Republic, peter.girovsky@tuke.sk