UDK 621.311

MODELLING PI CONTROLLED INDUCTION MOTOR BY USING ARTIFICIAL NEURAL NETWORKS

USTUN S.V., DEMIRTAS M.

Dept. of Electrical & Electronic Engineering
Celal Bayar University
Manisa, Turkey
e-mail:svustun@hotmail.com

Dept. of Electrical & Electronic Engineering
Dicle University
Diyarbakır, Turkey
e-mail: mdtas@dicle.edu.tr

ABSTRACT- This study presents system modelling based on obtaining real data from system for providing control requirements of nonlinear system. Back-propagation of error algorithm is used as learning algorithm for the Artificial Neural Network (ANN). The system model is obtained from the real system and controller by using ANN. ANN algorithm is realized to obtain maximum overshoot and settling time at the end of the learning process. The system recognizations with ANN process are realized for different K_p-K_i pairs taken from application circuit by using maximum overshoots and settling time. TMS320C50 is used as a digital signal processor. The implementation of ANN is described, and results are presented. The results presented show that ANN method not only has better dynamic performance but also has a less steady-state error.

1. INTRODUCTION

ANNs are recently showing good promise for application in power electronics and motion control systems. In the past few years, ANN has been used in some power electronic applications, such as inverter current regulation, dc motor control, flux estimation [1], and observer-based control of induction machines [2]. Evidently, ANN technique is showing promise as a competitive method of signal processing for power electronics applications. ANNs have the advantages of extremely fast parallel computation, immunity from input harmonic ripple, and fault tolerance characteristics due to distributed network intelligence.

Although ANNs have been around since the late 1950's, it wasn't until the mid-1980's that algorithms became sophisticated enough for general applications. Today ANNs are being applied to an increasing number of real- world problems of considerable complexity. They are good pattern recognition engines and robust classifiers, with the ability to generalize in making decisions about imprecise input data. They offer ideal solutions to a variety of classification problems such as speech, character and signal recognition, as well as functional prediction and system modelling where the physical processes are not understood or are highly complex. ANNs may also be applied to control problems, where the input variables are measurements used to drive an output actuator, and the network learns the control function. The advantage of ANNs lies in their resilience against distortions in the input data and their capability of learning. They are often good at solving problems that are too complex for conventional technologies (e.g., problems that do not have an algorithmic solution or for which an algorithmic solution is too complex to be found) and are often well suited to problems that people are good at solving, but for which traditional methods are not [3].

ANNs are successfully used in a lot of areas such as control, early detection of electric machine faults, digital signal processing in our daily technology [2]. A back-propagation training algorithm first proposed by Rumelhart, Hinton, and Williams in 1986 usually trains the feed-forward neural network. The distributed weights in the network contribute to the distributed intelligence or associative memory property of the network. With the network initially untrained, i.e., with the weights selected at random, the output signal pattern will totally mismatch the desired output pattern for a given input pattern. The actual output pattern is compared with the desired output pattern and the supervised back-propagation training algorithm adjusts the weights until the pattern matching occurs, i.e., the pattern errors become acceptably small [3].

In this study, identification of the system via ANN is done by using maximum overshoot and settling time obtained from the application circuit for different K_p-K_i pairs. In the application circuit, obtaining performance criteria and sending it to PC are Digital Signal Processor (DSP) based. Using TMS320C50 microprocessor of Texas Instrument makes the DSP application. In ANN and DSP applications, C++ and Assembler are used for ANN and DSP, respectively.

2. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Network is a system loosely modelled on the human brain. The field goes by many names, such as connectionism, parallel distributed processing, neuro-computing, natural intelligent systems, machine learning algorithms, and artificial neural networks. It is an attempt to simulate within specialized hardware or sophisticated software, the multiple layers of simple processing elements called neurons. Each neuron is linked to certain of its neighbours with varying coefficients of connectivity that represent the strengths of these connections. Learning is accomplished by adjusting these strengths to cause the overall network to output appropriate results [3].

The most basic components of neural networks are modelled after the structure of the brain. Some neural network structures are not closely to the brain and some does not have a biological counterpart in the brain. However, neural networks have a strong similarity to the biological brain and therefore a great deal of the terminology is borrowed from neuroscience.

Artificial Neural Networks are relatively crude electronic models based on the neural structure of the brain. The brain basically learns from experience. It is natural proof that some problems that are beyond the scope of current computers are indeed solvable by small energy efficient packages. This brain modelling also promises a less technical way to develop machine solutions. This new approach to computing also provides a more graceful degradation during system overload than its more traditional counterparts [4].

A neural network is a powerful data-modelling tool that is able to capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain.

There are multitudes of different types of ANNs. Some of the more popular include the multi-layer perceptron which is generally trained with the backpropagation of error algorithm, learning vector quantization, radial basis function, Hopfield, and Kohonen, to name a few. Some ANNs are classified as feed-forward while others are recurrent (i.e., implement feedback) depending on how data is processed through the network. Another way of classifying ANN types is by their method of learning (or training), as some ANNs employ supervised training while others are referred to as unsupervised or self-organizing. Supervised training is analogous to a student guided by an instructor. Unsupervised algorithms essentially perform clustering of the data into similar groups based on the measured attributes or features serving as inputs to the algorithms. The most common neural network model is the multi-layer perceptron (MLP). This type of neural network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown.

The training of MLP can be done with using backpropagation algorithm. With backpropagation, the input data is repeatedly presented to the neural network. With each presentation the output of the neural network is compared to the desired output and an error is computed. This error is then fed back (backpropagated) to the neural network and used to adjust the weights such that the error decreases with each iteration and the neural model gets closer and closer to producing the desired output. This process is known as "training".

Multi-layer perceptrons (MLPs) are the simplest and therefore most commonly used neural network architectures. Backpropagation algorithm is the most commonly adopted MLP training algorithm [5]. It is a gradient descent algorithm and gives the change $\Delta w_{ji}(k)$ in the weight of a connection between neurons i and j as follows

$$\Delta w_{ji}(k) = \eta \delta_j x_i + \alpha \Delta w_{ji}(k-1)$$

Where η is a parameter called the learning coefficient, α is the momentum coefficient, and δ_j is a factor depending on whether neuron j is an output neuron or a hidden neuron [6].

3. TRAINING AN ARTIFICIAL NEURAL NETWORK

Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then, the training, or learning, begins.

There are two approaches to training-supervised and unsupervised. Supervised training involves a mechanism of providing the network with the desired output either by manually "grading" the network's performance or by providing the desired outputs with the inputs. Unsupervised training is where the network has to make sense of the inputs without outside help [7].

The vast bulk of networks utilize supervised training. Unsupervised training is used to perform some initial characterization on inputs. However, in the full-blown sense of being truly self-learning, it is still just a shining promise that is not fully understood, does not completely work, and thus is relegated to the lab [3].

4. SYSTEM MODELLING WITH ANN

Used ANN model is multi-layer perceptron model in which there are more layer from one between input and output layer. Backpropogation of error algorithm is used as training algorithm. Backpropogation of error algorithm is used for training generalized delta rule. The training process of this ANN model is shown Fig. 1.

Obtaining of the system model with ANN consist of four stage.

1. Obtaining of input-output data of the system.
2. Selection of the ANN structure
3. Realizing of the training process
4. Testing of appropriateness of the ANN model

30 number input-output data taken from application circuit is given Table 1. ANN structure for this system is shown in Fig. 2. Also ANN parameters that model system are presented Table 2.

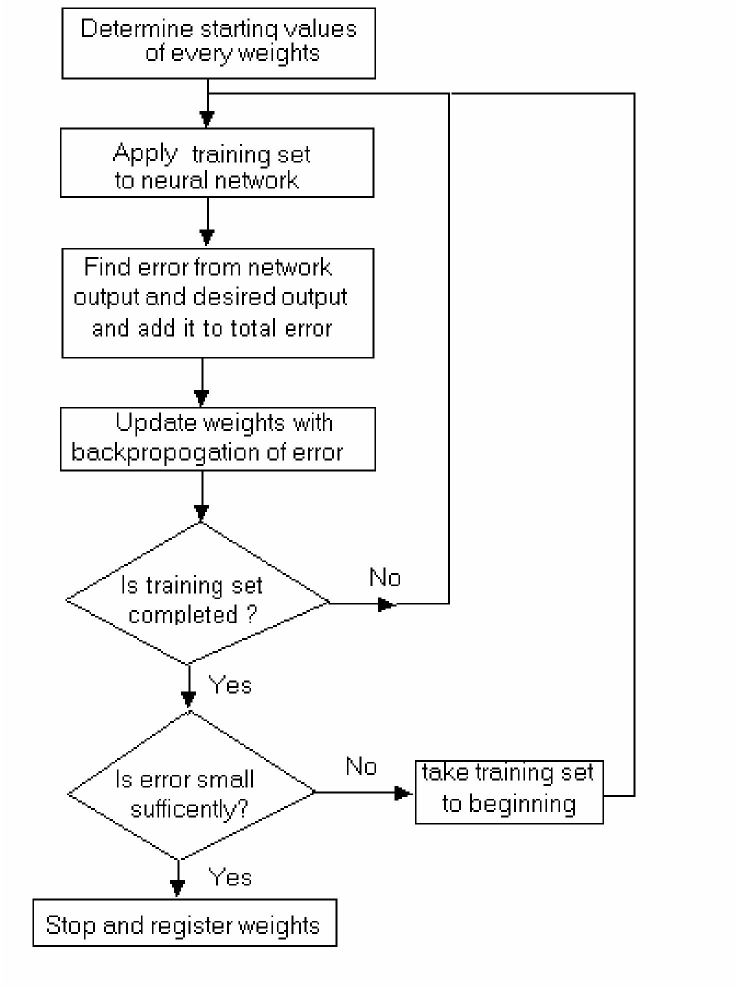

Fig. 1 The flow diagram of training process.

Table 1. Data used for training of ANN

	Kp	Ki	Maxsimum overshoot (rpm)	Settling time (sn)		Kp	Ki	Maxsimum overshoot (rpm)	Settling time (sn)
1	3,199	0,132	110,715	0,850	16	7,938	0,529	73,810	0,606
2	3,199	0,264	99,014	1,421	17	0,821	0,198	114,316	2,850
3	3,199	0,397	90,013	2,831	18	0,821	0,331	99,914	2,858
4	3,199	0,529	79,211	2,858	19	0,821	0,463	93,613	2,831
5	4,771	0,132	103,514	0,781	20	0,821	0,595	83,712	2,858
6	4,771	0,264	90,013	0,690	21	0,821	0,066	144,920	1,859
7	4,771	0,397	86,412	0,941	22	2,401	0,198	109,815	1,330
8	4,771	0,529	78,311	1,665	23	2,401	0,331	97,214	2,774
9	6,352	0,132	99,914	0,568	24	2,401	0,463	88,212	2,858
10	6,352	0,264	84,612	0,659	25	2,401	0,595	82,812	2,858
11	6,352	0,397	80,111	0,735	26	2,401	0,066	137,719	1,097
12	6,352	0,529	72,910	0,850	27	3,981	0,198	99,914	0,743
13	7,938	0,132	88,212	0,572	28	3,981	0,331	83,712	0,987
14	7,938	0,264	76,511	0,499	29	3,981	0,463	78,311	2,198
15	7,938	0,397	72,910	0,610	30	3,981	0,595	80,111	2,858

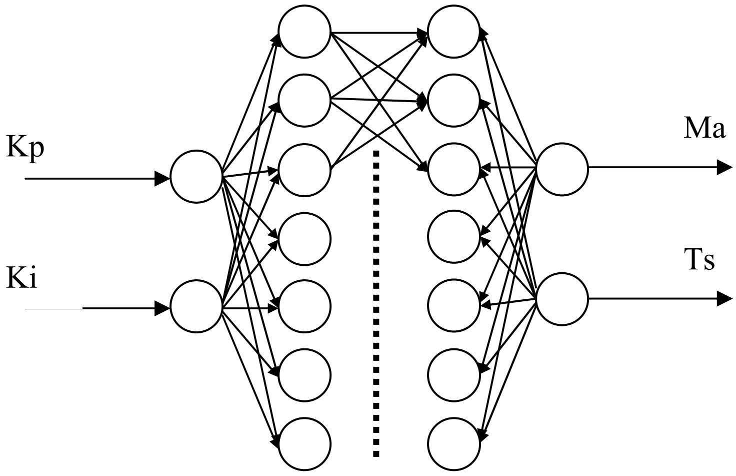

Fig. 2 ANN model structure of the system.

Table 2. ANN parameters of the modelling system

Number of neurons of the input layer	2
Number of neurons of the output layer	2
Layer number	2
First layer cell number	7
Second layer cell number	7
First layer activation function	Sigmoid
Second layer activation function	Sigmoid
Maxsimum iteration number	15000
Error limit	0,0001
Training coefficient	0,7
Momentum coefficient	0,9

There is not any criterion to select cell number at every layer number of the ANN structure. Layer number and cell number are determined with experiment. At the same way, teaching and momentum coefficients are determined based on experiences at previous study [5].

5. EXPERIMENTAL SETUP

We use motor and generator that is connected to motor with a connecting element. Motor was used 0.55 kW, 2.6A, 220V, 50Hz, Cosφ=0.79, 3-phase squirrel-cage induction motor. The processor used in this work is 40Mhz TMS320C50 Digital Signal Processor (DSP) with 10k x 16 words of on-chip RAM that works parallel with TLC320C40 Analogue interface circuitry (AIC) with 14-bit resolution. The processor is communicated with a PC through RS232 serial port. Block diagram of this application circuit is shown in Fig.3. Inverter is designed with Intelligent Power Module (IPM). DSP sends IGBT's ON-OFF data to IPM using an interface circuit. Trigger signals of Isolated-Gate-Bipolar-Transistor (IGBT) founded on the IPM are sent over the DSP data line and address line as synchronic. Thus, trigger of IGBT is provided. 150-300V, 8.5A, 1-2 kW, 1500-3000 rpm, UErr=220V, IErr=0.6A DC Motor which has a tachogenerator is used as a load. The stator voltage is adjusted by using Pulse Width Modulation (PWM) technique. After a period is modulated by 15-triangle wave, on-off data and duration are determined, and a table is built by using a C++ program. PI decides to which voltage packet is sent in the next step, according to the velocity error.

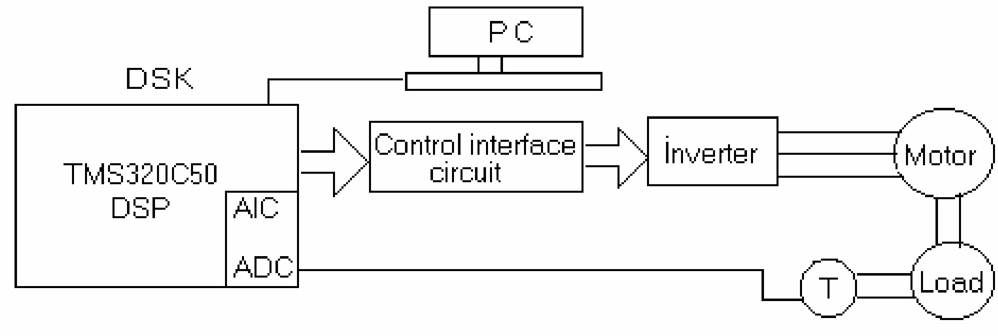

Fig. 3 Block diagram of the application circuit.

Analog/Digital (A/D) Interface on TMS320C50 DSP kit sends to the processor over the serial line, processing speed data come from tachometer connected the load, for forming control loop on the whole system to be control. Such as, A/D, D/A, filter,.. etc processes are realized as 14 bit. DSP system used on the control system is planned as RISC architecture speciality for high-speed Input/Output (I/O) comments. Because of the high-speed RISC comment set is selected 50 ns.

Realizing program of the control process is written in TMS320C50 assemble language. Controlling and compiling processes are made by compiler program. Program is sent to the processor from RS232 serial line via kernel program. IGBT operation situations are formed as ON-OFF data by using PWM technique. This is formed as table by using C++ language. Maximum pattern frequency is 19.2 KHz.

6. DISCUSSION

A part of the training data and change of the error at training process are shown in Fig. 4. Mean-squared error reduces lower of 0.001 at 1000 iterations. But iteration is continued to 15000 iterations. The training process is finished when mean-squared error reduces to 0.000330 at 15000 iterations.

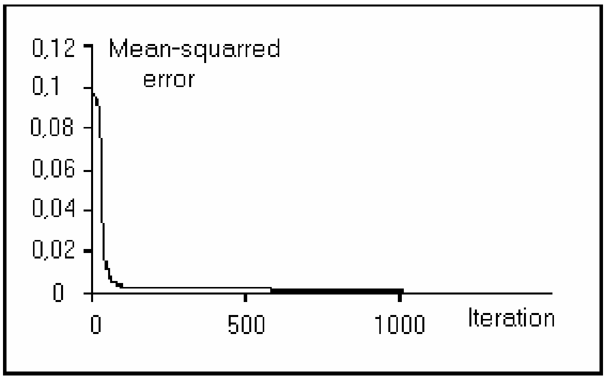

Fig. 4 Mean-squared error values according to iteration

Maximum overshoot (Ma) and settling time (Ts) are obtained from application circuit for different Kp and Ki pairs. These data are used for training of ANN.
Obtained model is tested with data that is not on the training set for appropriatness of the ANN model after the training process. Changing of settling time according to K_i for actual system and ANN model is shown in Fig. 5, where K_p is 2,828 and constant. Changing of settling time is shown in Fig. 6 for K_p=4,266, where there is too small error between ANN model and actual system output. Changing of the maximum overshoot value of the speed for actual system model and ANN model are demonstrated in Fig. 7 and 8 for K_p=2,828 and K_P=4,266, respectively, where ANN model follows system output with a small error. The error arises from situations that are realized as experimental and not to be modelled in

nonlinear system. Changing of maximum overshoot is 91-84-87-78,... when the results take one after another by using the same K_p-K_i coefficients. This special situation is shown in Fig.9.

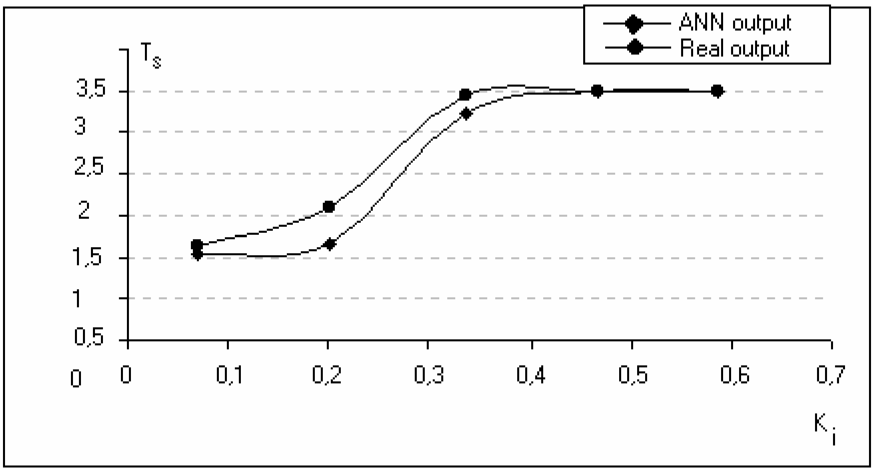

Fig. 5 Change of settling time (Ts) obtained from ANN and actual system according to K_i for K_p=2,828

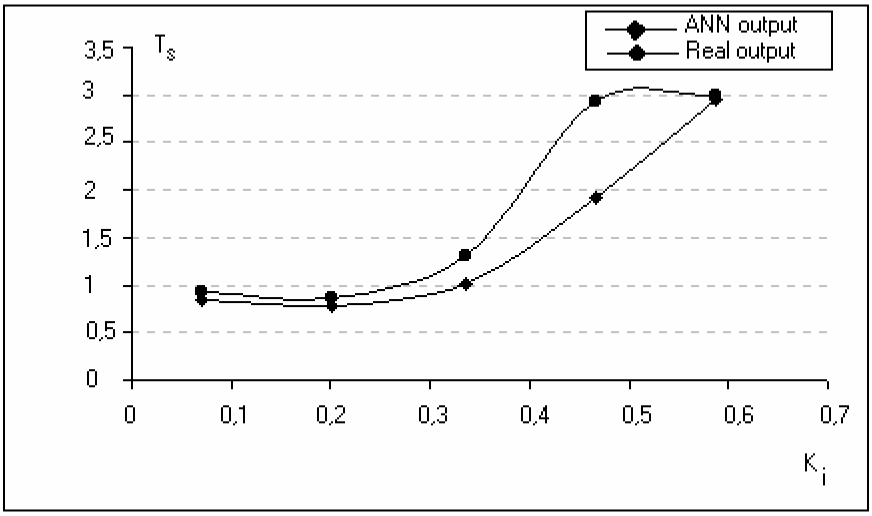

Fig. 6 Change of settling time (Ts) obtained from ANN and actual system according to K_i for K_p=4,266

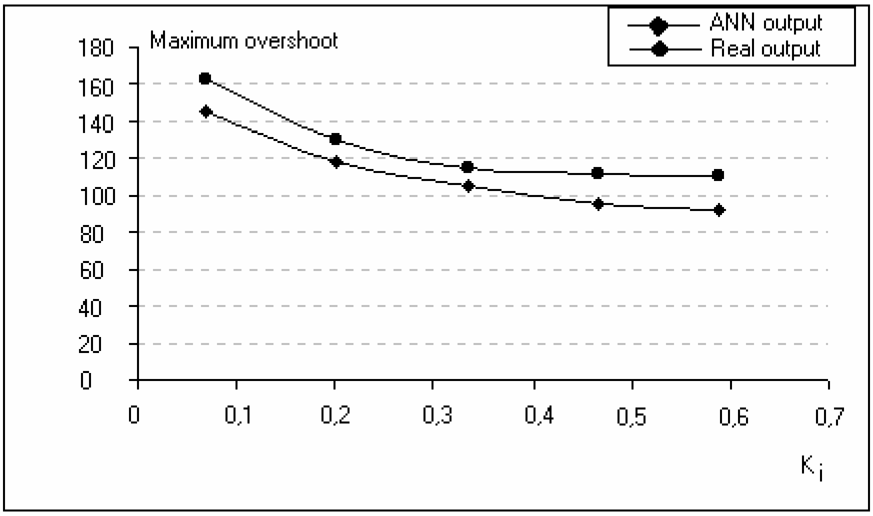

Fig. 7 Change of maximum overshoot obtained from actual system and ANN according to K_i for K_p=2,828

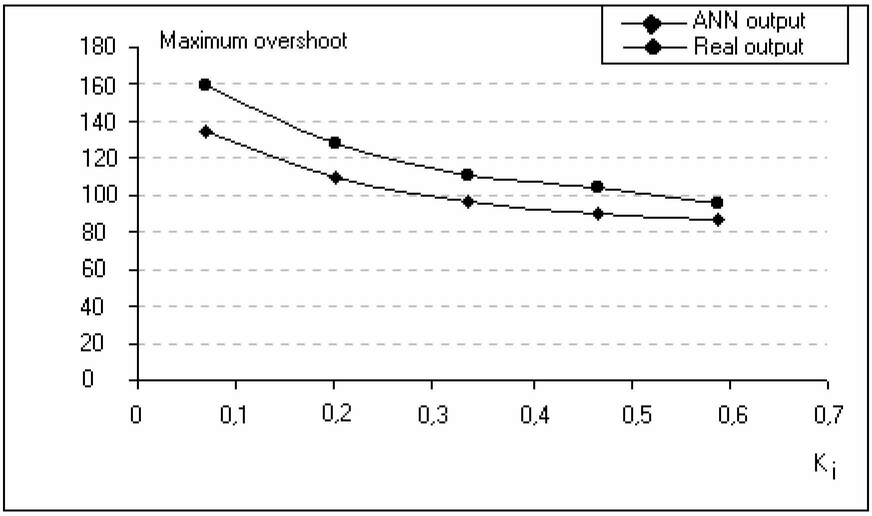

Fig. 8 Change of maximum overshoot obtained from actual system and ANN according to K_i for K_p=4,266

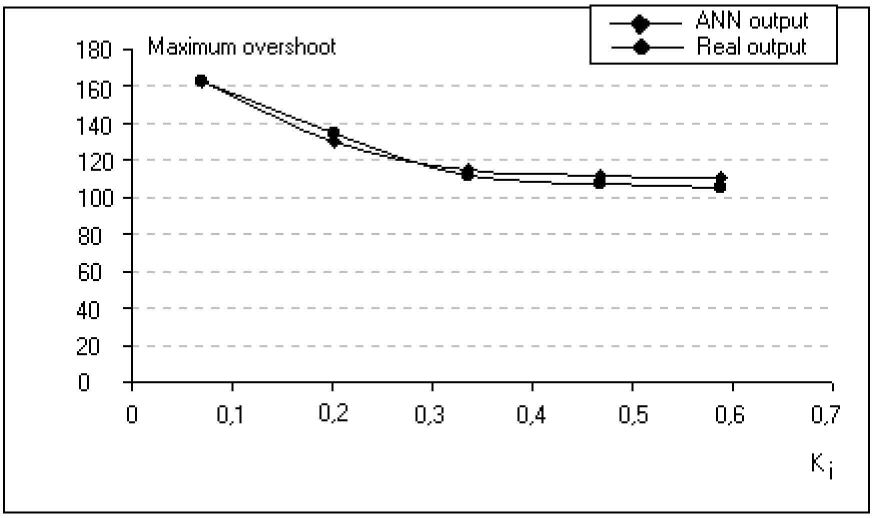

Fig. 9 The results obtained from actual system at the different time for the same K_p-K_i coefficients.

Obtained results are shown in Fig. 9, where the same values are taken two times for K_p=2,828 and increasing value of K_i. When ANN is trained with changing data, it is unavoidable to be a smaller error of the result. It is shown that ANN model formed for system models system successfully.

6. CONCLUSION

In this paper, actual system (motor and controller) is modelled by using ANN. Modelling of the induction motor is performed based on real data taken from the system for providing control requirements of it. Multi-layer perceptron model and backpropogation of error algorithm on the supervised training are used for modelling of the control system as ANN model.

Mean-squared error reduces less 0.001 at the 1000 iterations and 0.000330 at the 15000 iterations in which the training process is finished. Realizing program of the control process is written in TMS320C50 assemble language. This method is carried out as OFF-LINE. RISC comment set is selected 50 ns because I/O comments are high-speed.

It’s seen that ANN modelling can be represented the physical system exactly. The results presented show that ANN method has better dynamic performance. It’s obvious that, ANN successfully modelled for this system. This process can be also applied for nonlinear systems controlled PD and PID.

REFERENCES

[1] *A. K. P.Toh, E. P. Nowicki, and F. Ashafzaden,* “A Flux Estimator for Field Oriented Control of an Induction Motor Using an Artificial Neural Network”, IEEE Industry Applications Society Annual Meeting, pp. 585-592, 1994.

[2] *J. Theocharis and V. Petridis,* “Neural Network Observer for Induction Motor Control”, IEEE Contr. Syst. Mag., pp. 26-37, 1994.

[3] *Wassermann, Philip D.,* Neural Computing, Theory and Practice, Van Nostrand, NY, 1989.

[4] *Haykin Simon*, Neural Networks, Macmillan College Publishing Company Inc. ISBN 0-02-352761-7,1994.

[5*] S.V. Ustun,* “Optimisation of PI Coefficients for Controlling of Induction Motor Based on DSP by Using Artificial Neural Networks and Genetic Algorithms”, PhD. Thesis, Yildiz Technical University, Istanbul, 2001.

[6] *K.Guner, M.Erler and S.Sagiroglu,* “Artificial Neural Networks for the Characteristic Impedance Calculation of Conductor-Backed Coplanar Wave Guides”, Eleco’99 International Conference on Electrical and Electronics Engineering, 1999.

[7] *Kosko, Bart,* Neural Networks and Fuzzy Systems, Prentice Hall, Englewood Cliffs, NJ, 1992.

SÜNİ NEYRON ŞƏBƏKƏLƏRİNDƏN İSTİFADƏ ETMƏKLƏ PI KONTROLA MALİK İNDUKSİON ELEKTRİK MÜHƏRRİKLƏRİN MODELLƏŞDİRİLMƏSİ

ÜSTÜN S.V., DƏMİRTAŞ M.

İşdə sistemdən alınan həqiqi parametrlərə əsaslanaraq induksion elektrik mühərriklərinin modelləşdirilməsi yerinə yetirilmişdir. TMS320C50 markalı rəqəmli prosessordan istifadə edilmişdir. Alınmış nəticələrdən məlum olur ki, süni neyron şəbəkəsi üsulu yaxşı dinamik xarakteristikaya malik olmaqla yanaşı eyni zamanda dayanığlı rejimin xətasına malikdir.

МОДЕЛИРОВАНИЕ PI КОНТРОЛИРУЕМОГО ИНДУКЦИОННОГО ЭЛЕКТРОДВИГАТЕЛЯ С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

УСТУН С.В., ДЕМИРТАШ М.

В работе, на основе реальных параметров системы, проведено моделирование индукционного электродвигателя. Использовался цифровой процессор марки TMS320C50. Из полученных результатов выявлено, что метод искусственных нейронных сетей обладает не только лучшими динамическими характеристиками, но и меньшей погрешностью установившегося режима.