Paper Submission Form

Name of Presenting Author
Ahmet Bedri Özer

Address
Fırat Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği
Elazığ/Türkiye
Phone
(+90.424.2370000 (5292) cep: 0.533.3303642)
Fax
(+90.424. 2383787)
Other authors:
Erhan Akın

Title

ARTIFICIAL NEURAL NETWORK BASED SPEED ESTIMATOR FOR VECTOR CONTROLLED INDUCTION MOTOR

Related Topics (Please select from the Conference Home Page):

C6-A2

I would prefer presentation as (tick one)

( ) a poster **(/) an oral presentation** ( ) no preference

If we cannot accept your contribution in your preferred presentation mode, would you still be prepared
to present in the alternative mode (tick one):

(/) Yes ( ) No

ARTIFICIAL NEURAL NETWORK BASED SPEED ESTIMATOR FOR VECTOR CONTROLLED INDUCTION MOTOR

A. Bedri ÖZER **Erhan AKIN**
bozer@firat.edu.tr. eakin@firat.edu.tr.
Firat University, Department of Computer Engineering, 23119- ELAZIG/TURKEY

Index Terms: Artificial neural networks, backpropagation algorithm, induction motor, speed sensorless, vector control.

Abstract: In this paper, ANN-based (Artificial Neural Network-Based) algorithm estimates the rotor speed of vector controlled squirrel cage induction machine. In this study, stator currents, estimated stator voltages (which estimated from inverter's switching states) and a speed data that is taken from a speed sensor are used. For training, ANN-based estimator inputs are as selected: measured stator currents, estimated stator voltages and a past value of rotor speed. Output (target) is rotor speed, which is taken from a speed sensor. In the training, backpropagation training algorithm is used. Various test data is used and they give suitable results. In simulation results, it seems that proposed method is independent from parameter and load variations. The performance of the system is satisfactory.

I. INTRODUCTION

In the past, DC machines were commonly preferred. Because torque and speed controls of DC machines are easier than AC machines. In recent years, AC machines which are using vector control techniques have been developed. Nowadays, these techniques are more popular. In DC drives torque is proportional to armature current, so DC drive may directly control the torque by using a current control loop. Also flux control is easy in DC drives. The torque and flux controls are independent from each other. Because of these advantages, DC machines were popular in the past in spite of their high price and maintenance costs. Although the induction motors cost less, they had not got perfect torque and flux control until the vector control had been developed [1].

Vector controlled induction machine has replaced the DC motor, because it has achieved quick torque response. Requiring a speed sensor is the disadvantage of vector controlled induction machine. This speed sensor brings an extra cost and it reduces the system's reliability. So in recent years researchers have investigated speed sensorless systems [2-3].

Many techniques have been developed, but none of them has given perfect results in all conditions. There are many methods those have been improved to estimate the rotor speed. All of them have some advantages, but none of them has given perfect results. These methods should be grouped in two ways; conventional methods and artificial intelligence based methods. Conventional methods depend on parameters and variations of parameters. The reason of these variations may be skin effect, heat effect and etc. Online measurements of machine parameters are difficult (e.g. rotor resistance), so the systems that are less dependent on machine parameters are recommended.

In conventional methods, rotor speed is commonly calculated as difference between synchoronus angular velocity (ω_s) and slip angular velocity (ω_k) [4-5]. The fundamental of algorithm is given in equations (1-6);

$$\hat{\psi}_s = \int (u_s - i_s R_s) dt \qquad (1)$$

$$\hat{\psi}_s = [\psi_{sd}, \psi_{sq}]^T \qquad (2)$$

$$\theta = \tan^{-1}\left(\frac{\hat{\psi}_{sq}}{\hat{\psi}_{sd}}\right) \qquad (3)$$

$$\omega_s = \dot{\theta} = \frac{\hat{\psi}_{sd}\dot{\hat{\psi}}_{sq} - \hat{\psi}_{sq}\dot{\hat{\psi}}_{sd}}{\hat{\psi}_{sd}^2 + \hat{\psi}_{sq}^2} \qquad (4)$$

$$\omega_k = \frac{1}{\tau_r}\frac{i_{sq}}{i_{sd}} \qquad (5)$$

$$\omega_r = \omega_s - \omega_k \qquad (6)$$

τ_r is the rotor time constant.

These kinds of methods are depending on the parameters. Especially in low speeds, these techniques are given wrong results because the effect of voltage drop is descent and the value of stator resistance is variable. And also online measurement of stator resistance is difficult.

In another kinds of techniques, high frequency is enjected to stator and the speed is estimated using the effect of this frequency in rotor [6]. These techniques increase the cost of system.

In recent years, intelligent systems as, genetic algorithm, fuzzy logic, artificial neural network and hybrid of these methods (neuro-fuzzy or fuzzy-neuro) are popular and these systems may be applied to motor drive systems easily [7,8]. Conventional methods are mathematical model based systems, so it is difficult to control the behaviour of the system. But artificial intelligence based methods are easy to apply and less dependent on the machine parameters.

After various network structures and training algorithms have been improved, ANN have become very useful in induction motor drives [8,9]. ANN are very fast because of their parallel structure. They also have an advantage because of their error tolerance. If a neuron fails that will not affect the reliability of the system.

The advantages of ANN-based drive systems are as follows:

a) ANN are faster than other algorithms because of their parallel structure.
b) ANN do not require solution of any mathematical model.
c) If one neuron corrupts, the effect of this neuron is negligible.
d) If ANN are trained properly, they will give perfect results which will include all possible speeds and torque.
e) ANN are not dependent on the parameters, so the parameter variations do not effect the result.
f) ANN may be applied to other AC machines easily.

This paper presents ANN-based speed estimator, which is used in vector controlled squirrel cage induction motor. In second section there is a short view on the vector control of squirrel cage induction machine. The third section mentions the fundamentals of artificial neural networks. The fourth section expresses how the simulations are implemented and the results are also shown in the same section. The last section is the conclusion.

II. VECTOR CONTROL OF INDUCTION MOTOR

Induction machine has a complex structure. If three phases are transformed to two phases, the parameters that are non-linear and dependent on the time are eliminated [10].

$$
\begin{aligned}
u_{sd} &= R_s i_{sd} - \omega_s \psi_{sq} + p\psi_{sd} \\
u_{sq} &= R_s i_{sq} - \omega_s \psi_{sd} + p\psi_{sq} \\
0 &= R_r^{\cdot} i_{rd}^{\cdot} - \omega_k \psi_{rq}^{\cdot} + p\psi_{rd}^{\cdot} \\
0 &= R_r^{\cdot} i_{rq}^{\cdot} - \omega_k \psi_{rd}^{\cdot} + p\psi_{rq}^{\cdot}\,.
\end{aligned}
\tag{7}
$$

In this case, the solutions of equations are being less complex and easy to solve for processors (Equation (7)).

In these equations, the quantities of rotor are referred to stator quantities. The rotor voltages of squirrel cage induction motor are zero. If mechanical friction is neglected, the torque equation-11 is obtained.

$$T_e - T_y = \frac{2}{P} J \frac{d\omega_r}{dt} \tag{8}$$

As known, torque is obtained by product of armature voltage and airgap flux, which controlled by excited voltage, in DC machines.

$$T_e = K_e . \psi . I_e \tag{9}$$

If excited flux is constant then torque is controlled by armature current in DC machines. In AC machines, when stator current is changed, the components of stator current (i_{sq} and i_{sd}) are changed. And also rotor current components are changed too.

Figure-1 shows the vectoral quantities of induction machine in $\alpha - \beta$ and d-q frames. If d-axis and rotor flux are aligned then $\Psi_{rq} = 0$. But this alignment requires the position of rotor flux vector. Then d-q frame and rotor flux rotate together.

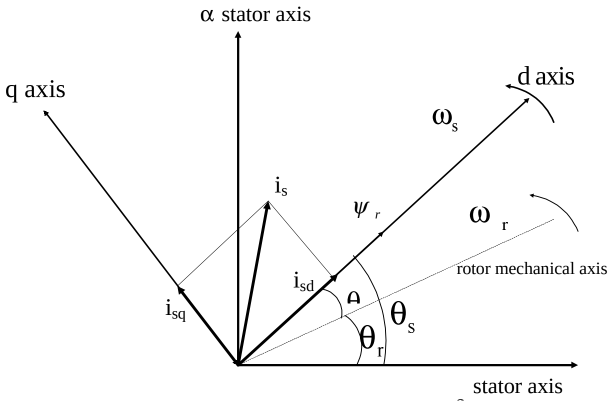

Figure-1 The vector diagram of induction machine

In vector control, the torque expression is as;

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m^2}{L_r^{\cdot}} i_{sd} i_{sq} \tag{10}$$

Now equation (9) and equation (10) become similar. So by vector control, the control of induction machine becomes easy like DC machine. But for validity of vector control, position of rotor flux (θ_s) must known.

III. ARTIFICIAL NEURAL NETWORKS

The human brain is a non-linear and complex structure. It behaves as a parallel computer. Brain performs certain computations many times faster than computers. Human brain has an excellent structure, it builds its own rules which we call experience. Artificial neural network is a system, which imitates the model of the human brain [11].

In general, ANN is a model of brain. In ANN, the neurons are connected to each other in various ways and the layers are frequently used. ANN are implemented either software or hardware.

ANN needs a learning process. In this process data is collected, these data is saved and generalised through weights. These weights are between neurons. To obtain the desired output, learning process includes learning algorithms, which achieve this by updating the weights.

Neurons are basic elements of ANN. The most common neuron model is shown in figure-3. A neuron consists of five parts. These parts are; inputs, weights, summing junction, activation function and outputs. Inputs take data from other neurons or from outside. Data joins the neuron by weights and these weights determine the effect of the relevant input. Summing junction calculates the net input, net input is the sum of the multiplication results of inputs and their relevant weights. Activation function applies the net input in a process and gives neuron output. Generally activation function is non-linear.

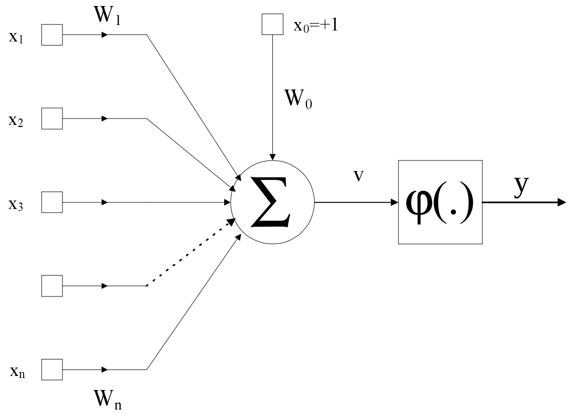

Figure-2 Artificial Neuron Model

x_0 which is shown in figure-2 is a constant. It is often called bias or threshold of the activation function. Mathematical model of a neuron is written as follows:

$$v = \sum_{i=0}^{n} w_i x_i \ , \quad y = \varphi(v) \tag{11}$$

Neurons are connected to each other in various ways and they create ANN. Neuron outputs can be connected to other neurons or to themselves by weights. And also these outputs can use delays. There are various kinds of ANN, which are separated from each other by their activation functions, learning rules or connection types.

The best property of ANN is that, they learn the system by using the actual examples. These actual examples are called training data. These training data is applied to the system and by the help of a learning rule, the system adjusts the weights by using output error. After training, different actual data, which is called test data, is applied to check the reliability of the system. A weight in the ANN is shown as;

$$W(k+1)=W(k) \pm \Delta W(k) \tag{12}$$

$\Delta W\ (k)$ is the correction coefficient, which is calculated by any learning rule in k.th time. Learning can be grouped in two ways; supervised and unsupervised.

Backpropagation algorithm is the most commonly used algorithm in supervised learning. This algorithm is simple and has a good learning capacity. Backpropagation algorithm uses two kinds of data: training data and test data. The training data is used to help the ANN to adjust the weights. By using the correct weights, the ANN is checked by testing data. If the ANN has not memorised the training data, then it should perform well on the test data and ANN can be applied to real problems. The selection of training data is very important. ANN give good performance only as good as the data used to train it. In selection of training data, it is not feasible to include every type of characteristic, because in this state ANN requires larger computational time. The ANN only requires data, which is suitable to learn the input-output relations. While training the network by backpropagation, the goal is to obtain a set of weights, such overall error is minimum. Overall error is the error between the desired output and actual output. ANN is trained by the sum of squared error.

$$e(k) = d(k) - y(k) \tag{13}$$

$$E(k) = \frac{1}{2} \sum_{l=1}^{p} e^2(k) \tag{14}$$

This algorithm minimizes the squared error between desired output and real output by using the gradient descent method.

There are two disadvantages of backpropagation algorithm; (i) slow approaches to result, (ii) to catch the local minimum. These disadvantages are disappeared by using learning rate and momentum coefficient.

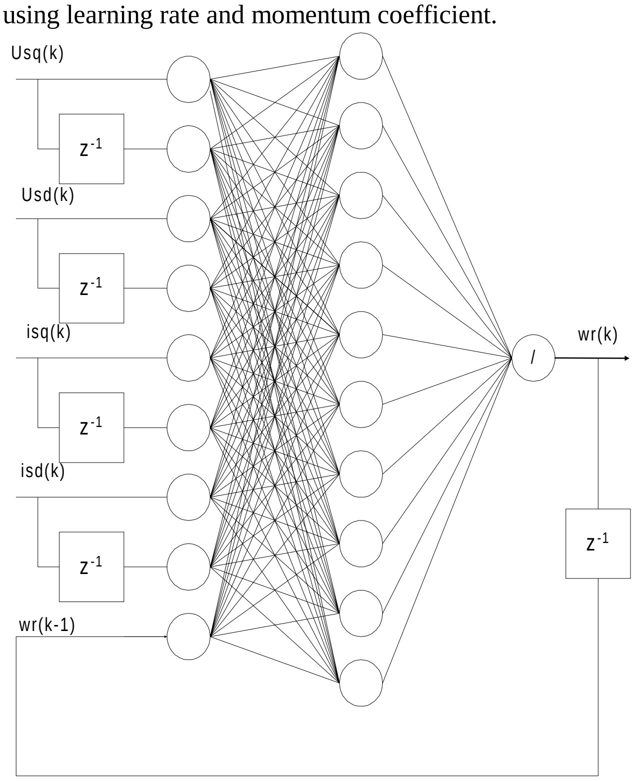

Figure-3 Proposed ANN-based speed estimator.

IV. PROPOSED ANN-BASED SPEED ESTIMATOR

A multilayer recursive ANN with the structure 8-9-1 (9 input nodes , 10 nodes in hidden layer, 1 output node) was obtained by trial and error to estimate the instantaneous rotor speed. In figure-3, ANN-based speed estimator structure has been showed. The inputs of ANN are the sampled values of stator currents, stator voltages and past value of the rotor speed (u_{sd} (k), u_{sd} (k-1), u_{sq} (k), u_{sq} (k-1), i_{sd} (k), i_{sd} (k-1), i_{sq} (k), i_{sq} (k-1), wr(k-1)). In proposed scheme a past values of stator voltages and stator currents are used inputs for good performance. Stator currents and voltages are in the stationary reference frame. The output of ANN is rotor speed (wr (k)). When the trained data was applied in checking phase, it gave correct results for those data as well. The activation function of input and hidden layer is tansigmoid. The experiment results showed that when tansigmoid activation function was used instead of logsigmoid activation function the training phase become shorter. The activation function of output layer is linear activation function. Training of ANN was performed with data corresponding the unloaded motor. But when a load was applied to the motor, ANN gave correct results. Activation function, number of hidden layers, number of neurons in each layer has been selected by trial and error. Backpropagation algorithm was used in proposed ANN. The input (voltages, currents) and output (rotor speed=target) of training data was given in figure-4. For training phase, all inputs and output were normalized between 0-1. Inputs were applied to the ANN and then error was obtained by difference between ANN's output and desired output. Backpropagation algorithm minimizes the error to desired value (sum squared error) by gradient descent method. In this algorithm weights were updated in training phase. In training phase 2800 data was used for each input and output (9 input + 1 output=10*2800=28000 data). Proposed ANN's sum squared error was selected 0.005. and after 168000 epochs the ANN achieved this target. Momentum coefficient was selected 0.95 and learning rate's default value was 0.00001 and it was adaptive.

In simulations, MATLAB 5.3, SIMULINK 3.0 and NNET 3.0 (Neural Network Toolbox) were used. The parameters of motor which was used in experiments are given in table-1. After training phase various test data was applied to the proposed ANN. Actual value, estimator output and the error (difference of actual value and estimator output) was given in figure-5. To show the system’s performance, the parameter value (stator resistance) was changed and also the motor was loaded. System was loaded to 1 Nm and stator resistance was changed to 5.5 Ω. After these changes, the proposed ANN-based estimator gave good results as shown in figure-6.

V. CONCLUSION

Finally, proposed ANN-based speed estimator is agreeable for high performance applications. The error between actual output and estimator output is trivial. And estimator estimates rotor speed truly when machine parameters are changed or motor is loaded. Requirement of speed sensor is disappeared. So proposed method is suitable for a drive applications.

Rated power	**1 hp**
Rated line voltage	**200 V**
Number of pole	**4**
Stator resistance (R_s)	**3.35 Ω**
Referred rotor resistance ($R_r^{'}$)	**1.99 Ω**
Stator leakage reactance (L_s)	**0.0024H**
Rotor leakage reactance ($L_r^{'}$)	**0.0024H**
Stator magnetizing reactance (L_m)	**0.0866H**
Rotor inertia	**0.05 kgm^2**

Table-1 Induction motor parameters

REFERENCES

[1]-Akın, E., 1994, Stator Akısı Üzerinden Asenkron Motorun Rotor Akısı Alan Yönlendirmesi için Bir Yöntem. Elektrik-Elektronik Anabilim Dalı Doktora Tezi, F.Ü. Fen Bilimleri Enstitüsü, Elazığ

[2]- Ohtani, T., Takada, N., Tanaka, K., 1992, Vector Control of Induction Motors without Shaft Encoder. IEEE Trans. on Ind. Applic.,Vol. 28, No. 1 February

[3]- Ilas, C., Bettini, A., Ferraris, L., Griva, G., Profumo, F., 1994, Comparison of Different Schemes without Shaft Sensors for Field Oriented Control Drives. IEEE

[4]-Özer, A.B., Can, H., Akın, E., 2000, Hız Duyargasız Vektör Kontrol Gerçekleştirilmesi. TOK’2000

[5]-Ben-brahim, L., Kawamura, A., 1992, A Fully Digitized Field-Oriented Controlled Induction Motor Drive Using Only Current Sensors, IEEE Trans. On Ind. Electr, Vol 39, No 3, June.

[6]-Holtz, J., 1998, Sensorless Position Control of Induction Motors - An Emerging Technology. IEEE Trans. on Ind Electr., Vol. 45, No. 6 December.

[7]- Rubai, A., Kankam, M.D., 2000, Adaptive Tracking Control of Induction Motor Drives Using Online Training of Neural Networks, IEEE Trans. On Ind. Aplic. Vol:39, No:5.

[8]-Vas, P., Artificial-Intelligence-Based Electrical Machines and Drives, Oxford University Press.

[9]- Rubai,A., Kotaru, R.,Online Identification and Control of Dc Motror using Learning Adaptation of Neural Networks, , IEEE Trans. On Ind. Aplic. Vol:36, No:3.

[10]-Ong, C.,1998, Dynamic Simulation of Electric Machinery (Using MATLAB/Simulink), Prentice Hill PTR.

[11]- Haykın, S., Neural Networks, A Comprehensive Foundation, Macmillan College Publishing Company,1994

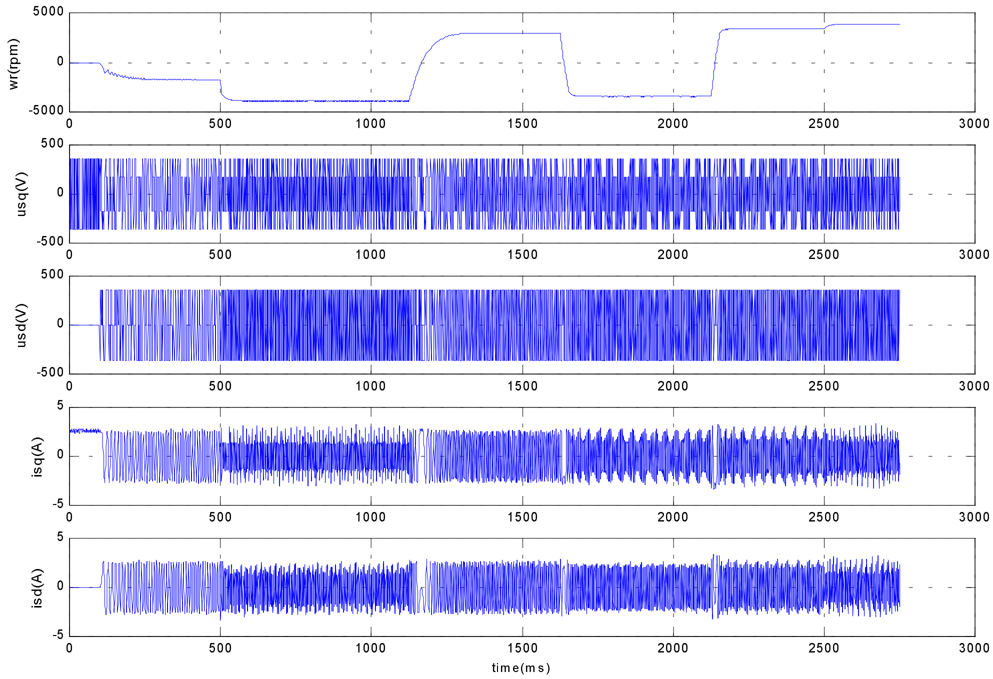

Figure-4 Training data (wr, u_{sq}, u_{sd}, i_{sq}, i_{sd})

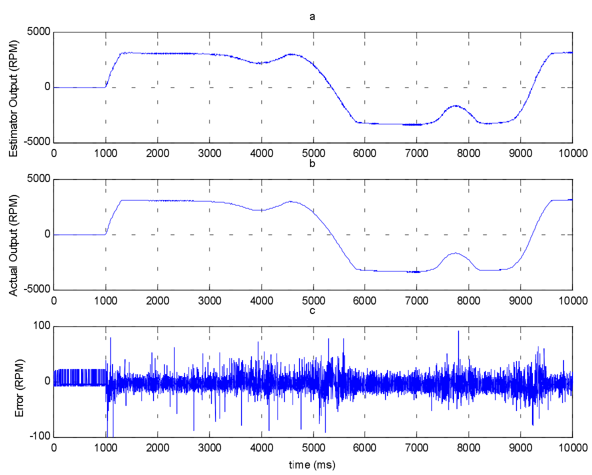

Figure-5 For any test data (a) ANN-Based Estimators output. (b) Actual output. (c) Difference between Estimators output and Actual output (error)

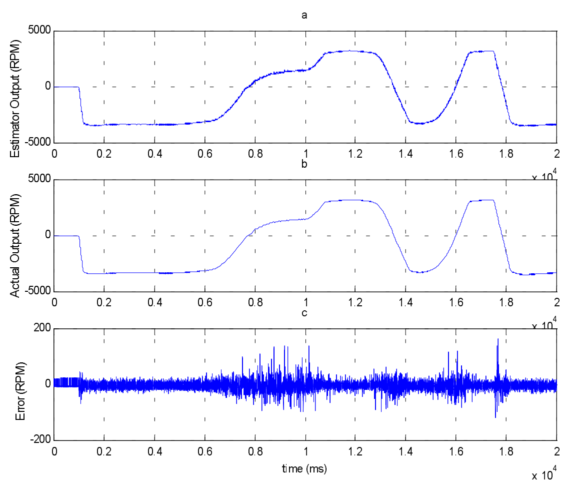

Figure-6 For any test data (stator resistance and load is different) (a) ANN-Based Estimators output. (b) Actual output. (c) Difference between Estimators output and Actual output (error)