36

Artificial Neural Network Applications in Power Electronics and Electrical Drives

B. Karanayil, Ph.D. and M.F. Rahman, Ph.D.
School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, New South Wales 2052, Australia

36.1 Introduction

In classical control systems, knowledge of the controlled system (plant) is required in the form of a set of algebraic and differential equations, which analytically relate inputs and outputs. However, these models can become complex, rely on many assumptions, may contain parameters which are difficult to measure or may change significantly during operation as in the case of the rotor flux oriented control (RFOC) induction motor drive. Classical control theory suffers from some limitations due to the assumptions made for the control system such as linearity, time-invariance, etc. These problems can be overcome by using artificial intelligence-based control techniques, and these techniques can be used, even when the analytical models are not known. Such control systems can also be less sensitive to parameter variation than classical control systems.

The main advantages of using artificial intelligence-based controllers and estimators are:

- Their design does not require a mathematical model of the plant.
- They can lead to improved performance, when properly tuned.
- They can be designed exclusively on the basis of linguistic information available from experts or by using clustering or other techniques.
- They may require less tuning effort than conventional controllers.
- They may be designed on the basis of data from a real system or a plant in the absence of necessary expert knowledge.
- They can be designed using a combination of linguistic and response-based information.

Generally, the following two types of intelligence-based systems are used for estimation and control of drives, namely:

(a) *Artificial Neural Networks (ANNs)*
(b) *Fuzzy Logic Systems (FLSs)*

In different applications in power electronics and electrical drives, there are occasions where an output y has to be estimated for an input x. This is generally accomplished with the help of mathematical equations of the system under consideration. Sometimes it may not be possible to have an accurate mathematical model, or there is no conventional model at all which can be used. In these circumstances, model-free

estimators required which can be either using ANNs or fuzzy logic systems. A mathematical model-based system can operate smoothly when there is no noise in the inputs but they might fail when the input has noise or if the inputs themselves are uncertain. Also, depending on the complexity of the mathematical model, the computation times can be excessive leading to difficulties for practical implementation.

A conventional function approximator, a neural function estimator, and how these estimators are arrived at are discussed briefly in Section 36.2. It will be shown how neural estimators are used in the estimation of speed, flux, torque, etc. for an induction motor. Some of these estimators are briefly reviewed in Section 36.3. Also, the ANNs have been used for identification and control of stator current in induction motor drives and these are briefly described in Section 36.4. In addition to their applications to motor drives, they are also used in control of power converters. Some of these applications are discussed in Section 36.5.

36.2 Conventional and Neural Function Approximators

36.2.1 Conventional Approximator

A conventional function approximator should give a good prediction of output data, when a system is presented with a new input data. A conventional function approximator uses a mathematical model of the system as shown in Fig. 36.1a. Sometimes it is not possible to have an accurate mathematical model of the system, in such a case; mathematical model free approximators are required, as shown in Fig. 36.1b. A conventional function approximator can easily be replaced by a neural network-based function approximator.

36.2.2 Neural Function Approximator

The function $y = f(x_1, x_2, \ldots, x_n)$ is the function to be approximated and $x_1, x_2, x_3, \ldots x_n$ are the n variables (n inputs) and the approximator uses the sum of non-linear functions $g_1, g_2, g_3, g_4, \ldots g_i$, where each of the g are non-linear functions of a single variable. Thus

$$y = f(x_1, x_2, \ldots, x_n) = g_1 + g_2 + g_3 + \cdots + g_{2n+1} = \sum_{i=1}^{2n+1} g_i \tag{36.1}$$

where g_i is a real and continuous non-linear function which depends only on a single variable z_i. The output y can also be represented as;

$$y = \sum_{j=1}^{n} w_{Mj} x_j = w_{M1} x_1 + w_{M2} x_2 + \cdots w_{Mn} x_n \tag{36.2}$$

These equations can be represented by a network shown in Fig. 36.2. There are n input nodes in the input layer, M hidden nodes in the hidden layer.

Figure 36.3 shows the schematic diagram of a neural approximation and Fig. 36.4 shows the technique of its training. The error (e) between the desired non-linear function (y) and the non-linear function ($\hat{y}$) obtained by the neural estimator

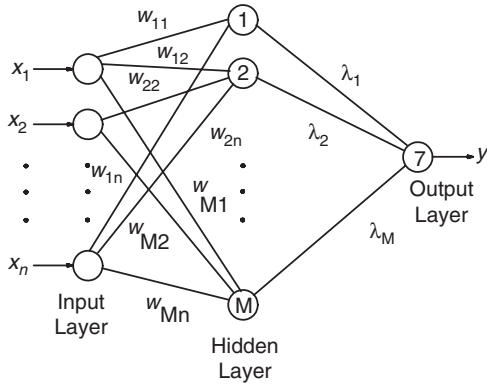

FIGURE 36.2 Artificial neural network to generate the required function.

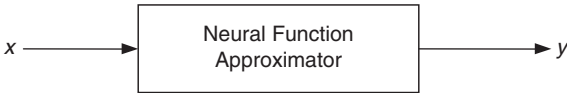

FIGURE 36.3 Schematic of a neural function approximator.

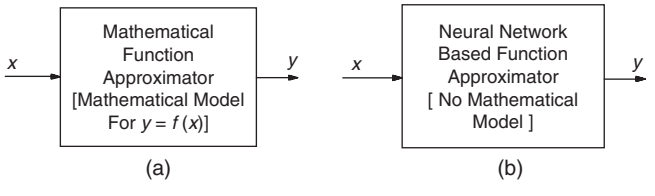

FIGURE 36.1 Conventional and neural network-based function approximators.

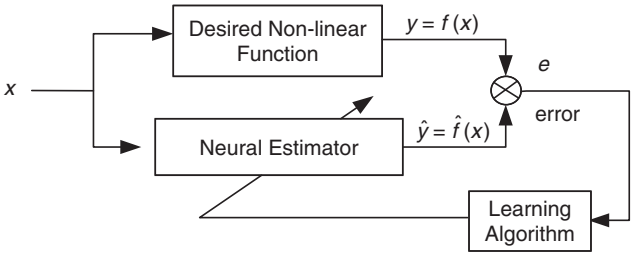

FIGURE 36.4 Training of a neural function approximator.

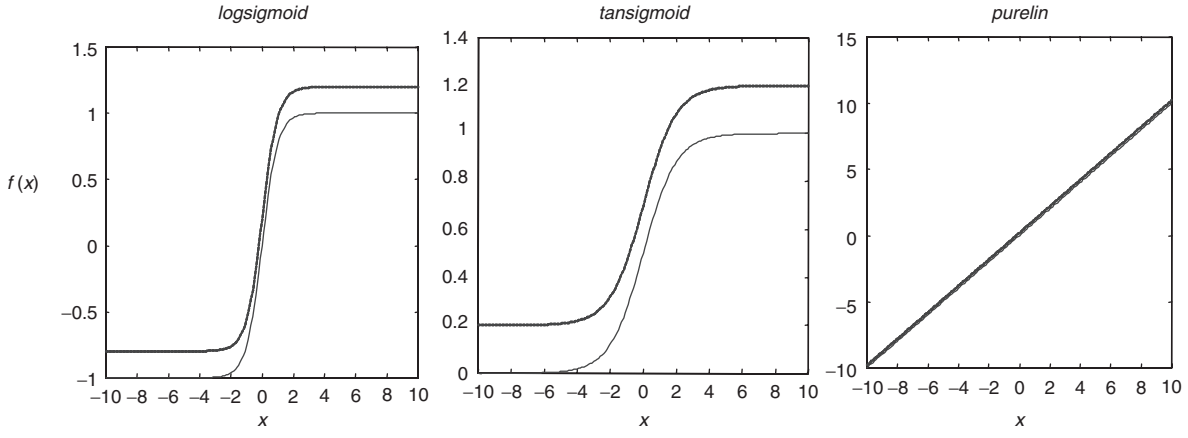

FIGURE 36.5 Logsigmoid, tansigmoid, and purelin activation functions.

is the input to the learning algorithm for the artificial neural estimator. This type of learning is known as back propagation.

The output of a single neuron can be represented as

$$a_i = f_i \left\{ \sum_{j=1}^{n} w_{ij} x_j(t) + b_i \right\} \tag{36.3}$$

where f_i is the activation function and b_i is the bias. Figure 36.5 shows a number of possible activation functions in a neuron. The simplest of all is the linear activation function, where the output varies linearly with the input but saturates at ± 1 as shown with a large magnitude of the input. The most commonly used activation functions are non-linear, continuously varying types between two asymptotic values 0 and 1 or -1 and $+1$. These are respectively, the sigmoidal function also called logsigmoid and the hyperbolic tan function also called tansigmoid.

The learning process of an ANN is based on the training process. One of the most widely used training techniques is the error back-propagation technique; a scheme which is illustrated in Fig. 36.4. When this technique is employed, the ANN is provided with input and output training data and the ANN configures its weights. The training process is then followed by supplying with the real input data and the ANN then produces the required output data.

The total network error (sum of squared errors) can be expressed as

$$E = \frac{1}{2} \sum_{k=1}^{P} \sum_{j=1}^{K} \left(d_{kj} - o_{kj} \right)^2 \tag{36.4}$$

where E is the total error, P is the number of patterns in the training data, k is the number of outputs in the network, d_{kj} is the target (desired) output for the pattern K and $o_{kj} (= y_{kj})$ is the jth output of the kth pattern. The minimization of the error can be arranged with different algorithms such as the gradient descent with momentum, Levenberg–Marquardt, reduced memory Levenberg–Marquardt, Bayesian regularization, etc.

36.3 ANN-based Estimation in Induction Motor Drives

Artificial neural networks have found widespread use in function approximation. It has been shown that, theoretically, a three layer ANN can approximate arbitrarily closely, any non-linear function, provided it is non-singular. Some of the ANN-based estimators reported in the literature for rotor flux, torque and rotor speed of induction motor drive are discussed in the following section.

36.3.1 Speed Estimation

In general, steady-state and transient analysis of induction motors is done using space vector theory, with the mathematical model having the parameters of the motor. To estimate the various machine quantities such as stator and rotor flux linkages, rotor speed, electromagnetic torque, etc., the above mathematical model is normally used. However, these machine quantities could be estimated without the mathematical model by using an ANN. Here no assumptions have to be made about any type of non-linearity.

As an example, the rotor speed of an induction motor can be estimated from the direct and quadrature axis stator voltages and currents in the stationary reference frame, as shown in Fig. 36.6. A three layer feedforward neural network structure with $8 \times 7 \times 1$ (8 input, 7 hidden, and 1 output) is used in this case. The input nodes were selected as equal to the number of input signals and the output nodes as equal to the number of output signals. The number of hidden layer neurons is generally taken as the mean of the input and output

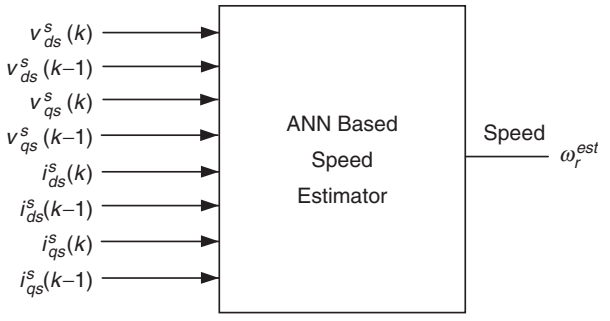

FIGURE 36.6 ANN-based speed estimator for induction motor.

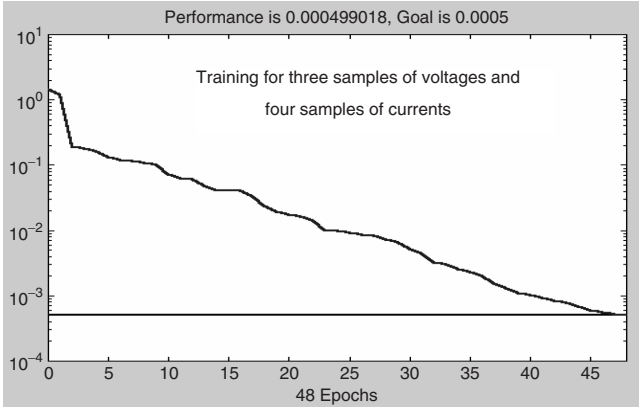

FIGURE 36.7 The error plot of $8 \times 7 \times 1$ network training for speed estimation.

nodes. In this speed estimator ANN, 7 hidden neurons were selected.

The results of an experiment conducted on a 1.1 kW, 415 V, 3 phase, 4 pole, 50 Hz induction motor is explained in this section. In order to investigate the case of speed estimation using ANN, a rotor flux oriented induction motor drive was set up in the laboratory, where the speed reference was changed in steps of 100 rpm and reversed every time the speed reached 1000 rpm. The load torque on the motor was kept constant at its full load rating. The stator voltages, stator currents, and the rotor speed were measured for 5 s and a data file was generated. The neural network was then trained using the *trainlm* algorithm with this data file. The training of the neural network converged after 48 epochs and the error plot for this network training is shown in Fig. 36.7. The estimated speed was predicted with the trained neural network, and the result is shown in Fig. 36.8.

The noisy data in the plot is the estimated speed and the continuous line is the speed measured with the encoder. The speed estimation was found to fail for speeds less than 100 rpm. If the trained neural network has to predict the speed under the complete range of operation of the drive, the data for training the neural network also has to be taken for the whole range. From this example investigated, it was found that the off-line training of the neural network could not produce satisfactory results, and it can be concluded that these off-line methods are not most suitable for these applications.

Artificial neural networks was also used for the estimation of the rotor speed of an induction motor together with the help of induction motor dynamic model. Though the technique gives a fairly good estimate of the speed, this technique lies more in the adaptive control area than in neural networks. The speed is not obtained at the output of a neural network; instead, the magnitude of one of the weights corresponds to the speed. The four quadrant operation of the drive was not possible for speeds less than 500 rpm. The motor was not able to follow the speed reference during the reversal for speeds less than 500 rpm. The drive worked satisfactorily for speeds above 500 rpm. Even though this method does not fall into a true neural network estimator, the results achieved with this type of implementation were very good except for lower speeds.

Alternately, the estimated speed can be made available at the output of a neural network as shown in Fig. 36.9. This speed estimator used a three layer neural network with five input nodes, one hidden layer, and one output layer to give the estimated speed $\hat{\omega}_r(k)$ as shown in Fig. 36.9. The three inputs to the ANN are a reference model flux λ_r^*, an adjustable model flux $\hat{\lambda}_r$, and $\hat{\omega}_r(k-1)$, the time delayed estimated speed. The multilayer and recurrent structure of the network makes it robust to parameter variations and system noise. The main advantage of their ANN structure lies in the fact that they have used a recurrent structure which is robust to parameter variations and system noise. These authors were able to achieve a speed control error of 0.6% for a reference speed of 10 rpm. The speed control error dropped to 0.584% for a reference speed of 1000 rpm.

36.3.2 Flux and Torque Estimation

The same principle as described in Section 36.3.1 can also be extended for simultaneous estimation of more quantities such as torque and stator flux. When more quantities or variables have to be estimated, the complex ANN has to implement a complex non-linear mapping.

The four feedback signals required for a direct field oriented induction motor drive can be estimated using ANNs. A $4 \times 20 \times 4$ multilayer network has been used for the estimation of the rotor flux magnitude, the electromagnetic torque, and the sine/cosine of the rotor flux angle. It has been demonstrated both by modeling and experimental results that the above estimated quantities were almost equal to the same quantities computed by a DSP-based estimator. Both the estimated torque and rotor flux signals using neural network was found to have higher ripple content compared to the DSP-based estimated quantities. It could be concluded that a properly trained ANN could totally eliminate the machine model equations as is evident from the results reported.

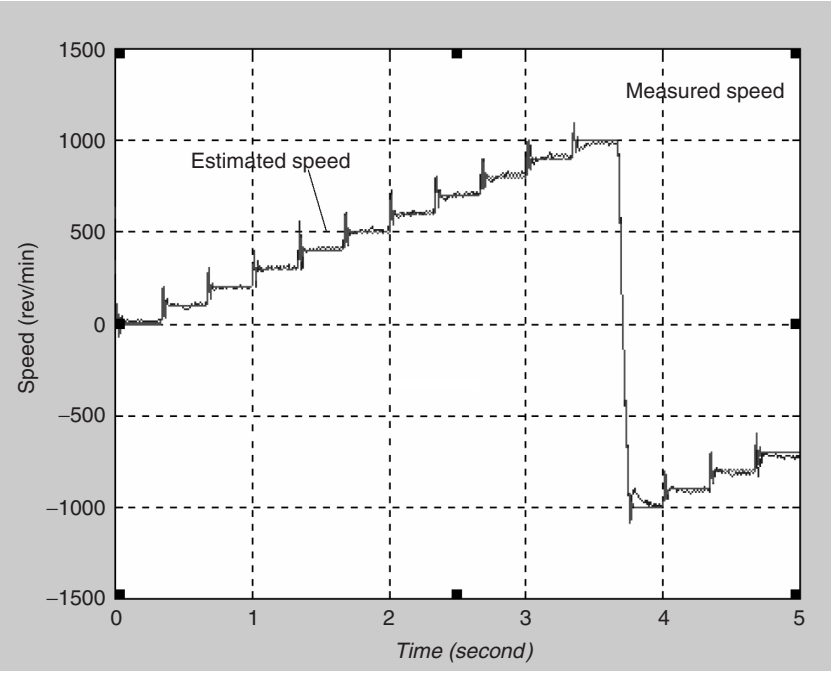

FIGURE 36.8 Measured speed vs estimated speed with ANN for induction motor.

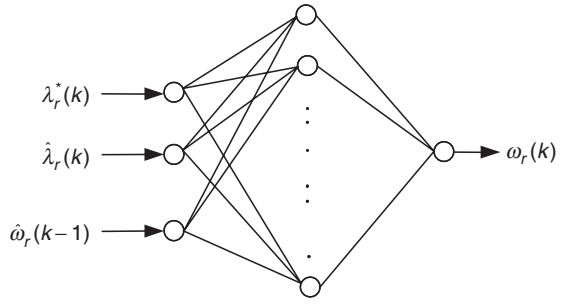

FIGURE 36.9 Structure of the neural network for the induction motor speed estimation.

In another application, an ANN with $5 \times 8 \times 8 \times 2$ structure has been used to estimate the stator flux using the measured induction motor stator variables quantities. After successful training, the ANN was used in a direct field oriented controlled drive. The rotor flux is computed from the stator flux estimate provided by the ANN and the stator current. This particular implementation also included an ANN-based decoupler which was used for the indirect field oriented (IFO) drive. An ANN with a structure of $2 \times 8 \times 8 \times 1$ was used for implementing the mapping between the flux and torque references and the stator current references. The estimated rotor flux using an ANN and a conventional FOC controller was shown to be equal. These authors have used experimental data for the ANN training and thus the effect of motor parameters was reduced. The authors were unable to use these estimated fluxes for controlling the induction motor in the experiment. The structure of the ANN used for this estimation is only that of a static ANN. It is preferable to have a dynamic neural network for this purpose.

36.3.3 Rotor Resistance Identification Using ANN

36.3.3.1 Off-line Trained ANN

The artificial neural network can also be used for the rotor time constant adaptation in indirect field oriented controlled drives. One of the implementation reported in the literature is shown in Fig. 36.10. There are five inputs to the T_r estimator, namely v_{ds}^s, v_{qs}^s, i_{ds}^s, i_{qs}^s, ω_r. The training signals are generated with step variations in rotor resistance for different torque reference T_e^* and flux command λ_r^* and the final network is connected in the IFO controller as shown in Fig. 36.10. The rotor time constant was tracked by a proportional integral (PI) regulator that corrects any errors in the slip calculator. The output of this regulator is summed with that of the slip calculator and the result constitutes the new slip command that is required to compensate for the rotor time constant variation. The major drawback of this scheme is that the final neural network is only an off-line trained neural network with a limited data file obtained from the modeling.

36.3.3.2 On-line Trained ANN

The limitations of off-line trained ANN are overcome by using an on-line trained ANN configuration. The method discussed

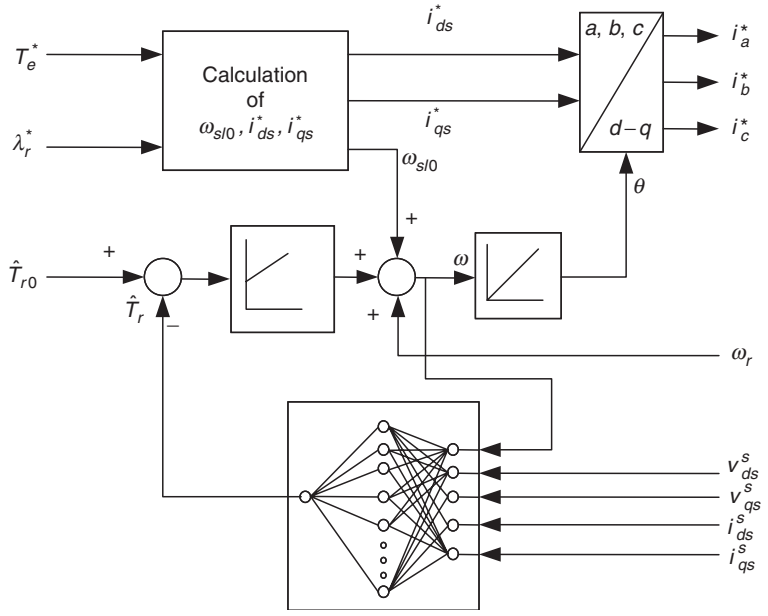

FIGURE 36.10 Principle of rotor time constant adaptation.

in this section has used an on-line trained ANN for adaptation of R_r of an induction motor in the RFOC induction motor drive. The error between the desired state variable of an induction motor and the actual state variable of a neural model is back propagated to adjust the weights of the neural model, so that the actual state variable tracks the desired value.

The principle of on-line estimation of rotor resistance (R_r) with multilayer feedforward artificial neural networks using on-line training has been described in Section 36.3.3.2.1. This technique was then investigated with the help of modeling studies with a 1.1 kW squirrel-cage induction motor (SCIM), described in Section 36.3.3.2.2. The modeling results are presented in Section 36.3.3.2.3. In order to validate the modeling studies, modeling results were compared with those from an experimental set-up with a SCIM under RFOC. These experimental results are presented in Section 36.3.3.2.4.

36.3.3.2.1 Multilayer Feedforward ANN Multilayer feedforward neural networks are regarded as universal approximations and have the capability to acquire non-linear input–output relationships of a system by learning via the back-propagation algorithm. It should be possible that a simple two-layer feedforward neural network trained by the back-propagation technique can be employed in the rotor resistance identification. The modified technique using ANN proposed in this section can be implemented in real time so that the resistance updates are available instantaneously and there is no convergence issues related to the learning algorithm. The two-layered neural network based on a back-propagation technique is used to estimate the rotor resistance. Two models of the state variable estimation are used, one provides the actual induction motor output and the other one gives the neural model output. The total error between the desired and actual state variables is then back propagated as shown in Fig. 36.11, to adjust the weights of the neural model, so that the output of this model coincides with the actual output. When the training is completed, the weights of the neural network should correspond to the parameters in the actual motor.

36.3.3.2.2 Rotor Resistance Estimation for RFOC Using ANN The basic structure of an adaptive scheme described by Fig. 36.11 is extended for rotor resistance estimation of an induction motor as illustrated in Fig. 36.12. Two independent observers are used to estimate the rotor flux vectors of the induction motor. If the rotor flux linkages are estimated using the stator voltages and stator currents, they are referred to as *voltage model* and if the rotor flux linkages are estimated using the stator currents and rotor speed they are referred to as *current model.*

The stator flux linkages based on the neural network model in Fig. 36.12 can be represented using Eq. (36.5), which is derived from the sample data models of the combined voltage and current model equations of the induction motor.

$$\overrightarrow{\lambda_r^{s^{nm}}}(k) = W_1X_1 + W_2X_2 + W_3X_3 \qquad (36.5)$$

The neural network model represented by Eq. (36.5) is shown in Fig. 36.13, where W_1, W_2, and W_3 represent the weights of the networks and X_1, X_2, X_3 are the three inputs to

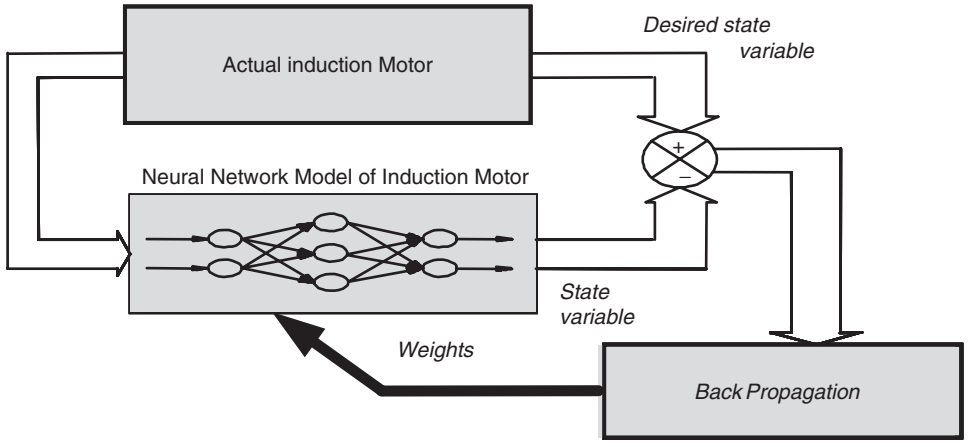

FIGURE 36.11 Block diagram of rotor resistance identification using neural networks.

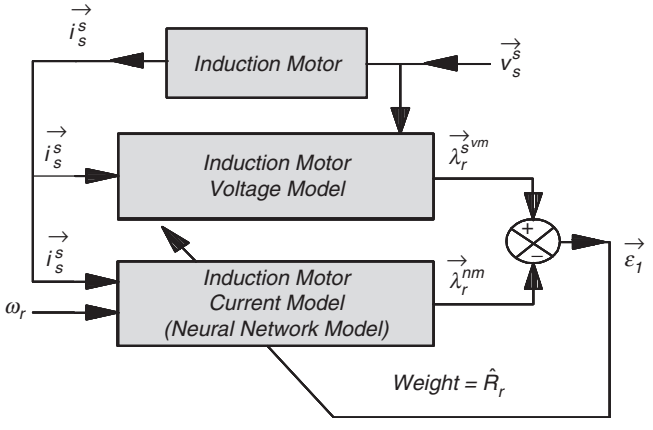

FIGURE 36.12 Structure of the neural network system for R_r estimation.

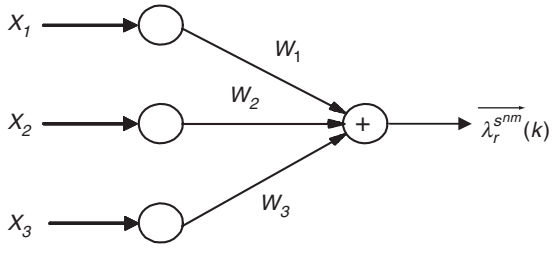

FIGURE 36.13 Two-layered neural network model for rotor flux linkage estimation.

the network. If the network shown in Fig. 36.13 has to be used to estimate R_r, where W_2 is already known, then W_1 and W_3 need to be updated.

The weights of the network, W_1 and W_3 are found from training, so as to minimize the cumulative error function E_1,

$$E_1 = \frac{1}{2}\vec{\varepsilon_1}^2(k) = \frac{1}{2}\left\{\overrightarrow{\lambda_r^{svm}}(k) - \overrightarrow{\lambda_r^{sim}}(k)\right\}^2 \tag{36.6}$$

The weight of the neural network W_1 has to be adjusted using generalized delta rule.

To accelerate the convergence of the error back-propagation learning algorithm, the current weight adjustment has to be supplemented with a fraction of the most recent weight adjustment, as indicated in Eq. (36.7).

$$W_1(k) = W_1(k-1) - \eta_1\vec{\delta}X_2 + \alpha_1\Delta W_1(k-1) \tag{36.7}$$

where α_1 is a user-selected positive momentum constant and η_1 is the training coefficient.

The rotor resistance R_r can now be calculated from W_3 from Eq. (36.8) as follows:

$$\hat{R}_r = \frac{L_r W_3}{L_m T_s} \tag{36.8}$$

36.3.3.2.3 Modeling Results A schematic diagram showing the implementation of a rotor resistance estimator in RFOC controller for induction motor is shown in Fig. 36.14. The stator voltages and currents are measured to estimate the rotor flux linkages using the voltage model as shown in this figure. The inputs to the rotor resistance estimator (RRE) are the stator currents, rotor flux linkages λ_{dr}^{svm}, λ_{qr}^{svm}, and the rotor speed ω_r. The estimated rotor resistance $\hat{R}_r$ will then be used in the RFOC controllers for the flux model. The response of the drive together with the rotor resistance estimator OFF is shown in Fig. 36.15 for an abrupt change in R_r of motor, from 6.03 to 8.5 Ω at 0.8 s. The possible changes in the estimated motor torque T_e, the rotor flux linkage λ_{rd} with this estimator were noted.

Subsequently the results of the drive were looked at with RRE ON, so that the rotor resistance in the controller R_r' was updated with the estimated rotor resistance $\hat{R}_r$ as shown in Fig. 36.16. The estimated rotor resistance $\hat{R}_r$ has converged to the rotor resistance of the motor R_r within 50 ms.

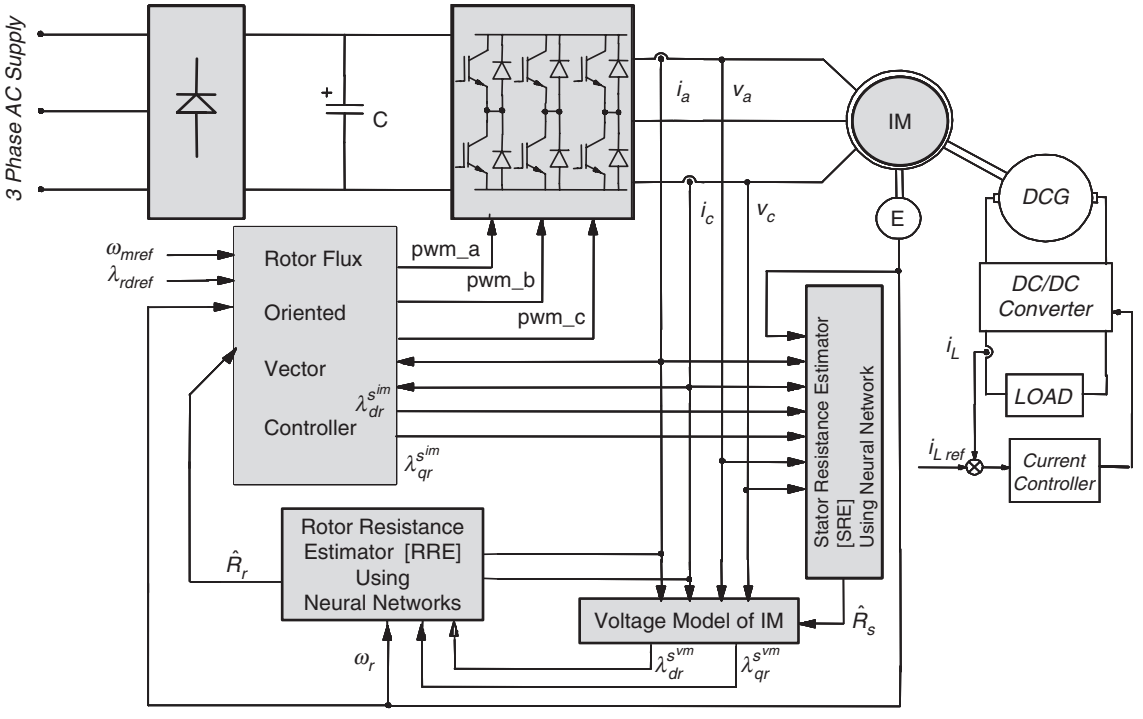

FIGURE 36.14 Block diagram of the RFOC induction motor drive with on-line rotor and stator resistance tracking using ANN.

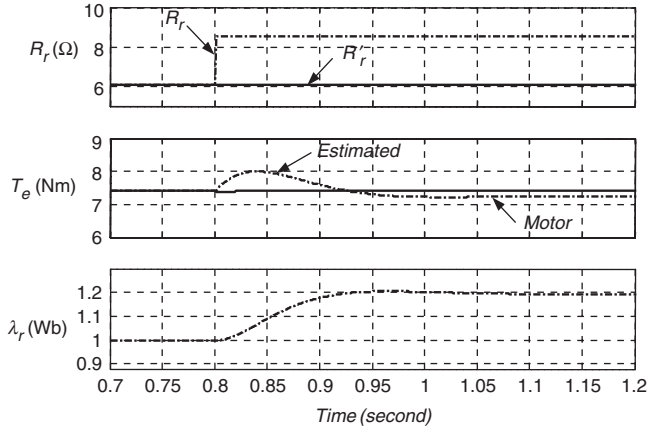

FIGURE 36.15 Effect of rotor resistance variation without rotor resistance estimator for 40% step change in R_r – modeling results.

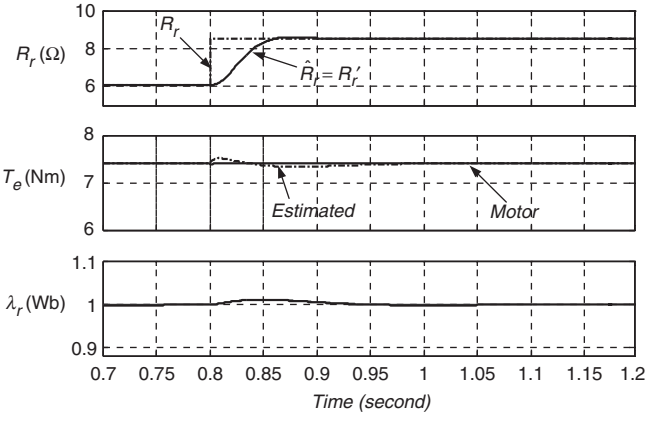

FIGURE 36.16 Effect of rotor resistance variation with rotor resistance estimator using ANN for 40% step change in R_r – modeling results.

The response of the drive system with the RRE using ANN are also shown for a practical profile in load torque, speed, and change in R_r. The R_r was increased from 6.03 to 8.5 Ω over a period of 8 s and the torque and rotor flux linkage are shown in Fig. 36.17 when the RRE is OFF. For rapid reversals of the drive and variations in load torque, there are significant errors in rotor flux linkages and thus errors in estimated torques. Subsequently, the RRE block was switched ON and it can be observed that the rotor resistance estimator was tracking very well even during the worst dynamics in the motor speed and load torques as shown in Fig. 36.18. The estimated rotor resistance $\hat{R}_r$ tracked the actual rotor resistance of the motor R_r throughout except there was some small error during the reversal of the motor. The rotor flux linkage λ_{rd} was found to remain constant at 1.0 Wb during this transient condition. The current i_{qs} also remained constant as the torque is now perfectly decoupled.

36.3.3.2.4 Experimental Results The practical implementation of the above rotor resistance estimation was found

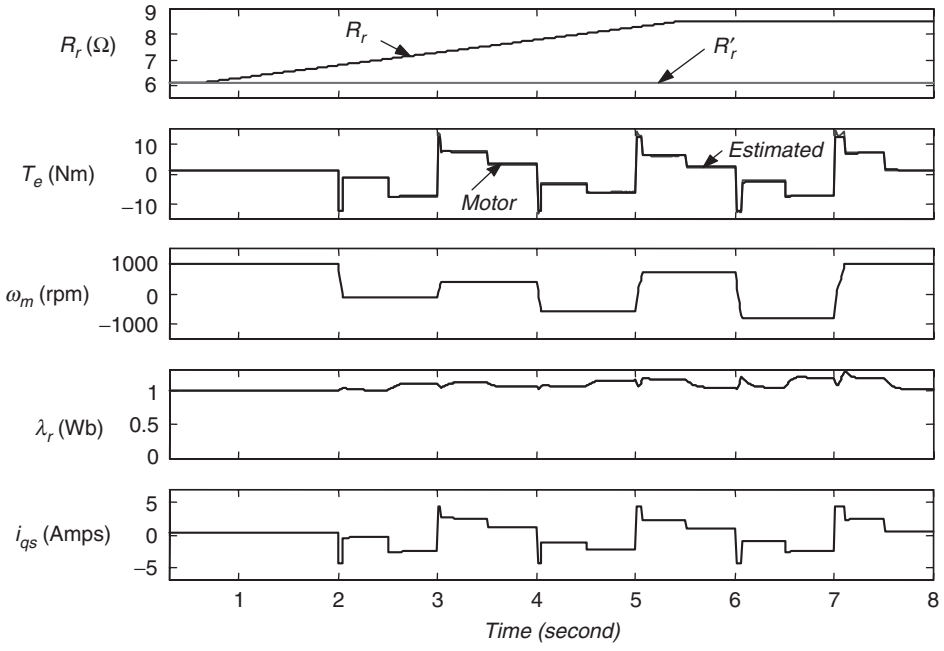

FIGURE 36.17 Effect of rotor resistance variation without rotor resistance estimator for 40% ramp change in R_r – modeling results.

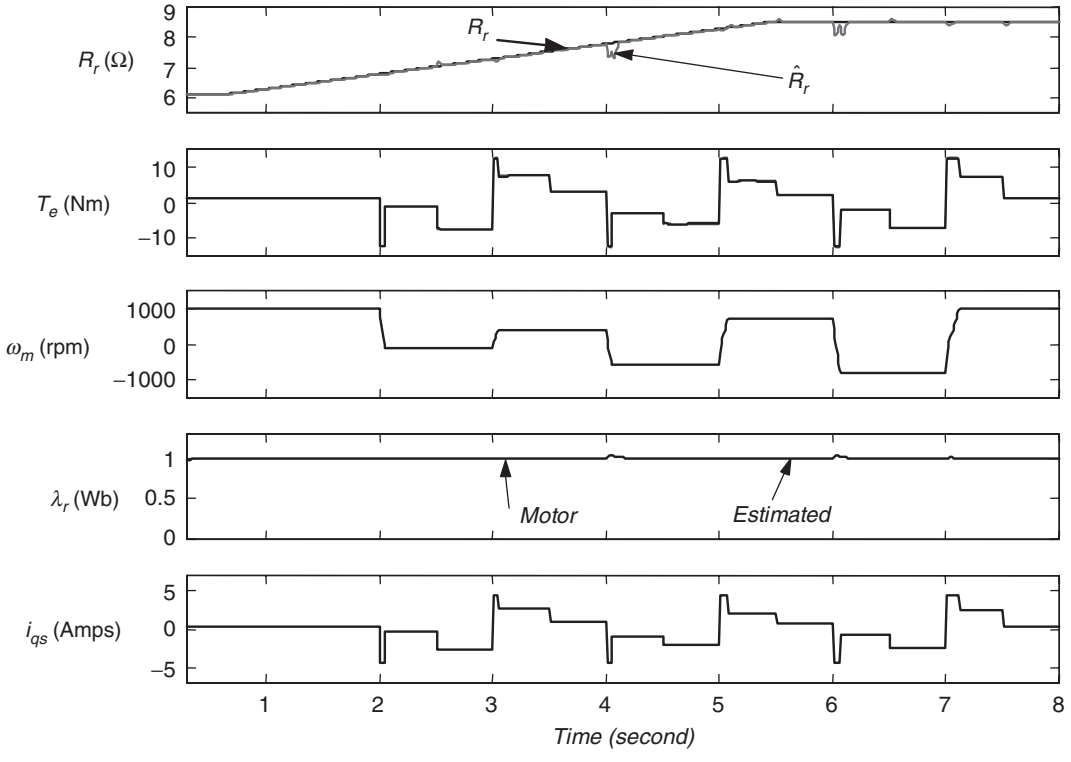

FIGURE 36.18 Effect of rotor resistance variation with rotor resistance estimator using ANN for 40% ramp change in R_r – modeling results.

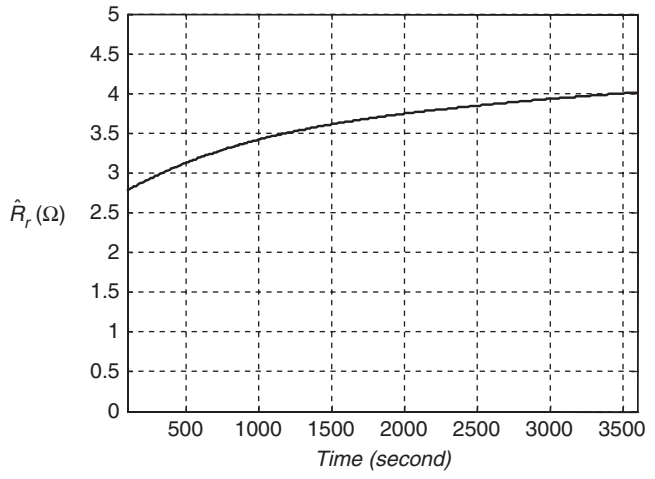

FIGURE 36.19 Estimated R_r using ANN – experimental results.

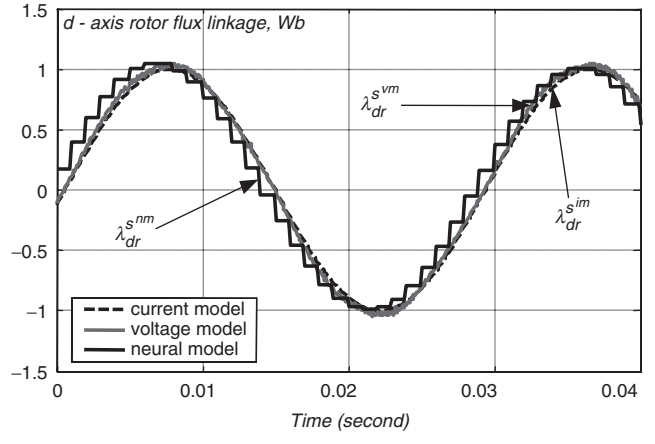

FIGURE 36.20 Rotor fluxes estimated during R_r estimation using ANN – experimental results.

functioning very well in the experimental drive set-up. The results of the R_r estimation obtained from the experiment is shown in Fig. 36.19 taken from a heat-run test conducted on an induction motor. As the RRE calculated, the rotor resistance using variables in stationary reference frame, the *d*-axis rotor flux linkages of the current model $\left(\lambda_{dr}^{im}\right)$, the voltage model $\left(\lambda_{dr}^{vm}\right)$, and the neural model $\left(\lambda_{dr}^{nm}\right)$, taken at the end of heat run are also recorded as shown in Fig. 36.20.

36.3.4 Stator Resistance Estimation Using ANN

In this section, the capability of a neural network has been deployed to have on-line estimator for stator resistance in an RFOC induction motor drive. The stator resistance observer was realized with a recurrent neural network with feedback loops trained using the standard back-propagation learning algorithm. Such architecture with recurrent neural network is known to be a more desirable approach and the implementation reported in this section confirms this.

The *d*-axis stator current in the stationary reference frame in the discrete form can be represented using the voltage and current model equations of the induction motor,

$$i_{ds}^{s*}(k) = W_4 i_{ds}^{s*}(k-1) + W_5 \lambda_{dr}^{s\,im}(k-1) + W_6 \omega_r \lambda_{qr}^{s\,im}(k-1) + W_7 v_{ds}^{s}(k-1) \tag{36.9}$$

where, $W_4 = 1 - \dfrac{T_s}{\sigma L_s}\dfrac{L_m^2}{L_r T_r} - \dfrac{T_s}{\sigma L_s}R_s$; $W_5 = \dfrac{T_s}{\sigma L_s}\dfrac{L_m}{L_r T_r}$

$W_6 = \dfrac{T_s}{\sigma L_s}\dfrac{L_m}{L_r}$; $W_7 = \dfrac{T_s}{\sigma L_s}$

The weights W_5, W_6, and W_7, are calculated using the induction motor parameters, rotor speed ω_r, and the sampling interval used in the estimator T_s.

The relationship between stator current and stator resistance is non-linear which could be easily mapped using a neural network.

When this Eq. (36.9) is represented graphically, it resembles a recurrent neural network as shown in Fig. 36.21. The standard back-propagation learning rule can then be employed for training this neural network. The weight W_4 is the result of training so as to minimize the cumulative error function E_2,

$$E_2 = \frac{1}{2}\vec{\varepsilon}_2{}^2(k) = \frac{1}{2}\left\{i_{ds}^{s}(k) - i_{ds}^{s*}(k)\right\}^2 \tag{36.10}$$

To accelerate the convergence of the error back-propagation learning algorithm, the current weight adjustment is supplemented with a fraction of the most recent weight adjustment, as indicated in Eq. (36.11).

$$W_4(k) = W_4(k-1) + \eta_2 \Delta W_4(k) + \alpha_2 \Delta W_4(k-1) \tag{36.11}$$

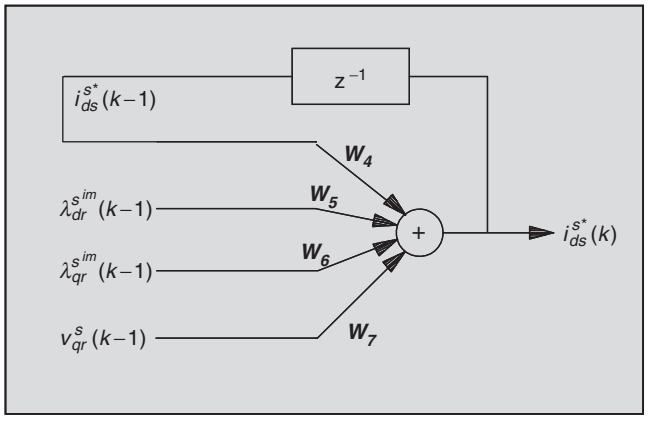

FIGURE 36.21 *d*-Axis stator current estimation using recurrent neural network based on Eq. (36.9).

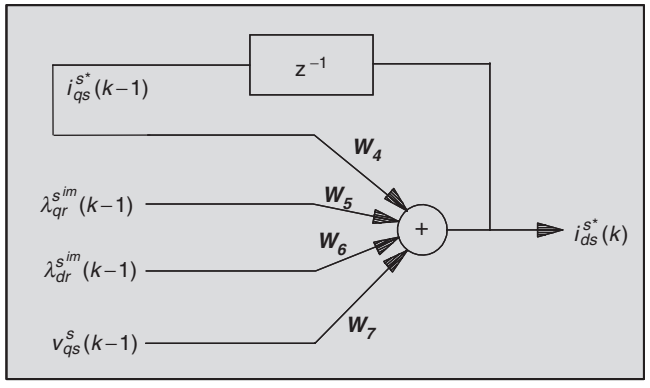

FIGURE 36.22 q-Axis stator current estimation using recurrent neural network based on Eq. (36.12).

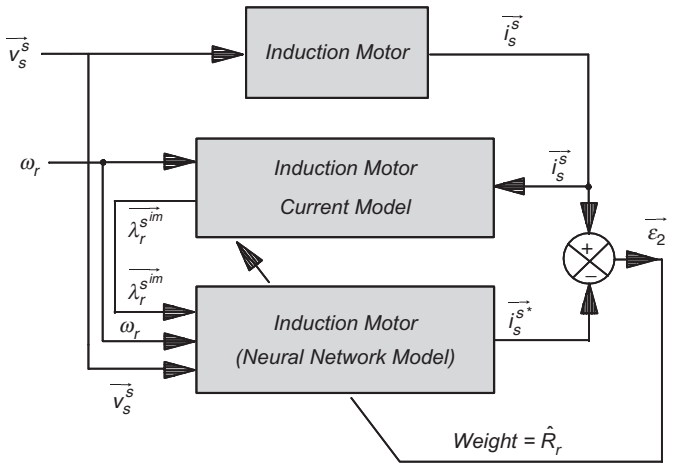

FIGURE 36.23 Block diagram of R_s estimation using artificial neural network.

where η_2 is the training coefficient, α_2 is a user-selected positive momentum constant.

Following a similar procedure, the q-axis stator current of the induction motor can be estimated using the discrete form as shown in Eq. (36.12),

$$i_{qs}^{s*}(k) = W_4 i_{qs}^{s*}(k-1) + W_5 \lambda_{qr}^{s\,im}(k-1) - \omega_r W_6 \lambda_{dr}^{s\,im}(k-1) + W_7 v_{qs}^{s}(k-1) \tag{36.12}$$

Equation (36.12) can be represented by a neural network as shown in Fig. 36.22. The weight W_4 is updated with the training based on Eq. (36.12).

The stator resistance $\hat{R}_s$ of the induction motor can now be calculated using Eq. (36.13) as follows:

$$\hat{R}_s = \left\{ 1 - W_4 - \frac{T_s}{\sigma L_s} \frac{L_m^2 \hat{R}_r}{L_r^2} \right\} \frac{\sigma L_s}{T_s} \tag{36.13}$$

The stator resistance of an induction motor can be thus estimated from the stator current using the neural network system as indicated in Fig. 36.23.

36.3.4.1 Modeling Results of Stator Resistance Estimation Using ANN

The block diagram of a rotor flux oriented induction motor drive together with stator resistance identification is already shown in Fig. 36.14, where the stator resistance estimation is implemented by the stator resistance estimator (SRE) block.

The stator resistance estimation results are as shown in Fig. 36.24. It has three results (1) without both rotor and stator resistance estimators, (2) with only rotor resistance estimation, and (3) with both rotor and stator resistance estimations. As shown in the figure, both R_r and R_s were increased abruptly by 40% at 1.5 s. It can be seen that the estimated stator resistance $\hat{R}_s$ converges to $\hat{R}_s$ within 200 ms. It can be noted from this figure that the convergence of the stator resistance estimation is not affected by the convergence of the rotor resistance estimator.

36.3.4.2 Experimental Results of Stator Resistance Estimation Using ANN

The stator resistance estimation algorithm was tested together with the rotor flux oriented induction motor drive of Fig. 36.14 implemented in the laboratory. To test the stator resistance estimation, an additional 3.4 Ω per phase was added in series with the induction motor stator, with the motor running at 1000 rev/min and with a load torque of 7.4 Nm.

The estimated stator resistance together with the actual stator resistance is shown in Fig. 36.25. The estimated stator resistance converges to 9.4 Ω within less than 200 ms.

Figure 36.26 shows both the measured d-axis stator current and the one estimated by the neural network model. The neural network model output $i_{ds}^{s*}(k)$ follows the measured values $i_{ds}^{s}(k)$, due to the on-line training of the neural network.

36.4 ANN-based Controls in Motor Drives

36.4.1 Induction Motor Current Control

Another application of ANN is to identify and control the stator current of an induction motor. In one study, Burton *et al.* have used a current control strategy outlined by Wishart and Harley to train an ANN to control the induction motor stator currents. They have used a training algorithm named random weight change (RWC) which is reported to be slightly faster than back-propagation. In RWC algorithm, the weights are perturbed by a fixed step-size and a random sign.

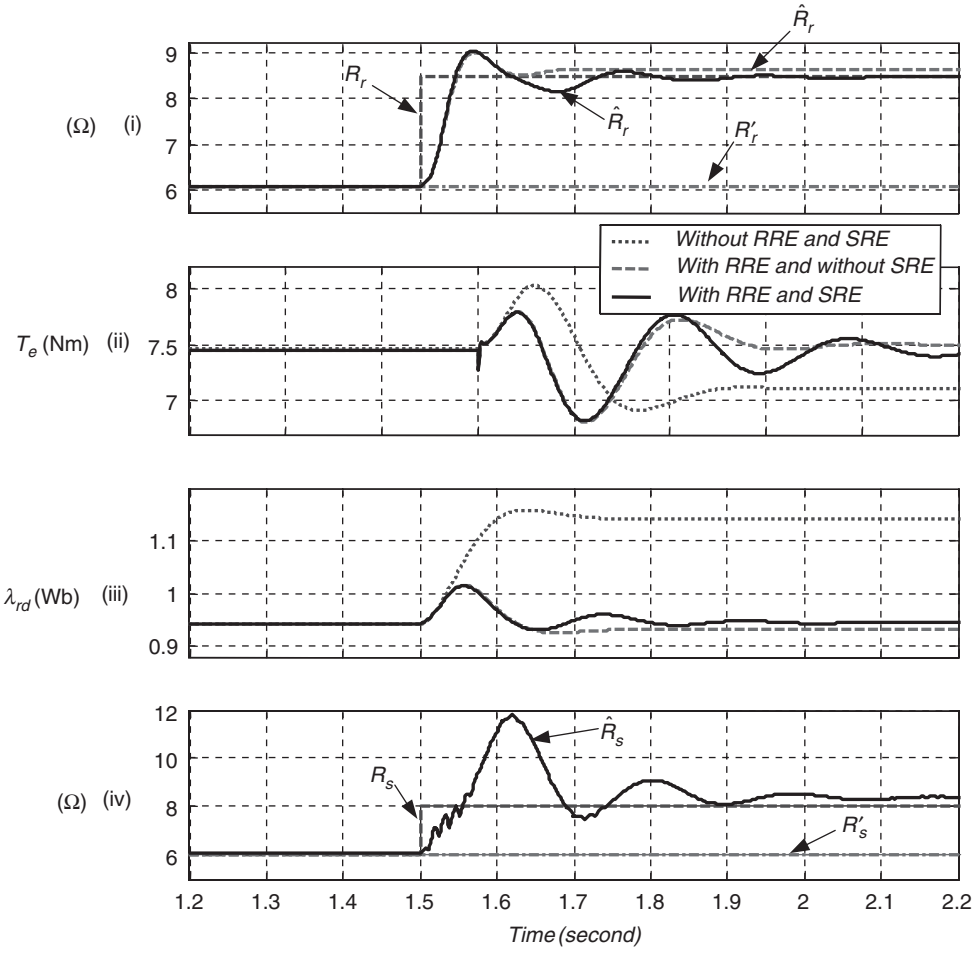

FIGURE 36.24 Performance of the drive with and without RRE and SRE using ANN for 40% step change in R_r and R_s, R'_r and R'_s compensated – modeling results.

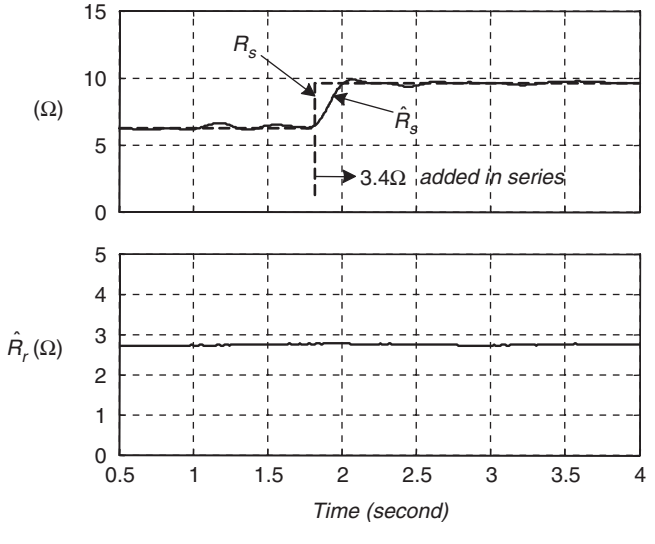

FIGURE 36.25 Estimated stator resistance R_s using ANN – experimental results.

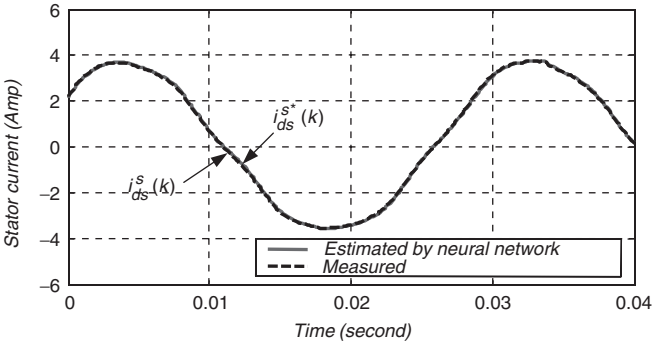

FIGURE 36.26 Stator currents in R_s estimation – experimental results.

This is done for fixed number of trials and after each trial, the error with the desired output is computed. Finally, the set of weight changes which result in the least error are chosen and the whole process is repeated till convergence is reached. The modeling results reported in this scheme was excellent. Later, Burton *et al.* presented their practical implementation

of this proposed stator current controller with a transputer controller card.

36.4.2 Induction Motor Control

Narendra and Parthasarathy proposed methods for identification and control of dynamical systems using ANNs. Wishart and Harley used the above basic principles to identify and control induction machines. A block diagram of the control scheme is shown in Fig. 36.27. For the induction motor, the Non-linear AutoRegressive Moving Average with eXogenous inputs (NARMAX) model for the stationary frame stator current is derived and used for the identification of electromagnetic model. In its general form, the NARMAX model represents a system in terms of its delayed inputs and outputs. Random steps in the stator voltage are given for the purpose of identification. The neural network used is of the multilayer back-propagation type, and a quantity based on the rotor time constant is also computed as an extra weight. As opposed to the regular ANN architecture, this ANN has non-linearity in the output layer and the weighted sum of the inputs is used as the output. This gives an estimate of the rotor time constant and makes the system robust against variation of the parameters. Once, the identification is over, the ANN is used for current control. The stator currents predicted by the ANN are used to compute the input voltage for the induction motor, and the ANN output is made to track the reference currents by back-propagating the error.

The rotor speed is also controlled in this system by identifying a NARMAX model for the speed increment rather than the absolute value of speed. To simplify the NARMAX model, the load torque is assumed to be a function of the motor speed, as is the case in a fan or pump type of load. For the current control case, the relationship between the control variable (voltage) and the controlled quantity (current) was linear. In the speed control case, this relationship is non-linear, thus necessitating two ANNs, one for identification of speed and the other for control.

The identification ANN (N_i) predicts the value for the speed increment, which is compared with the actual speed increment and the error (ε_i) is back-propagated through the ANN. A PI controller is used for basic speed control, and the control ANN (N_c) produces the slip frequency, and the difference between the desired speed increment and actual speed increment (ε_c) is back-propagated through the ANN. The induction motor drive therefore employs three ANNs as shown in Fig. 36.28.

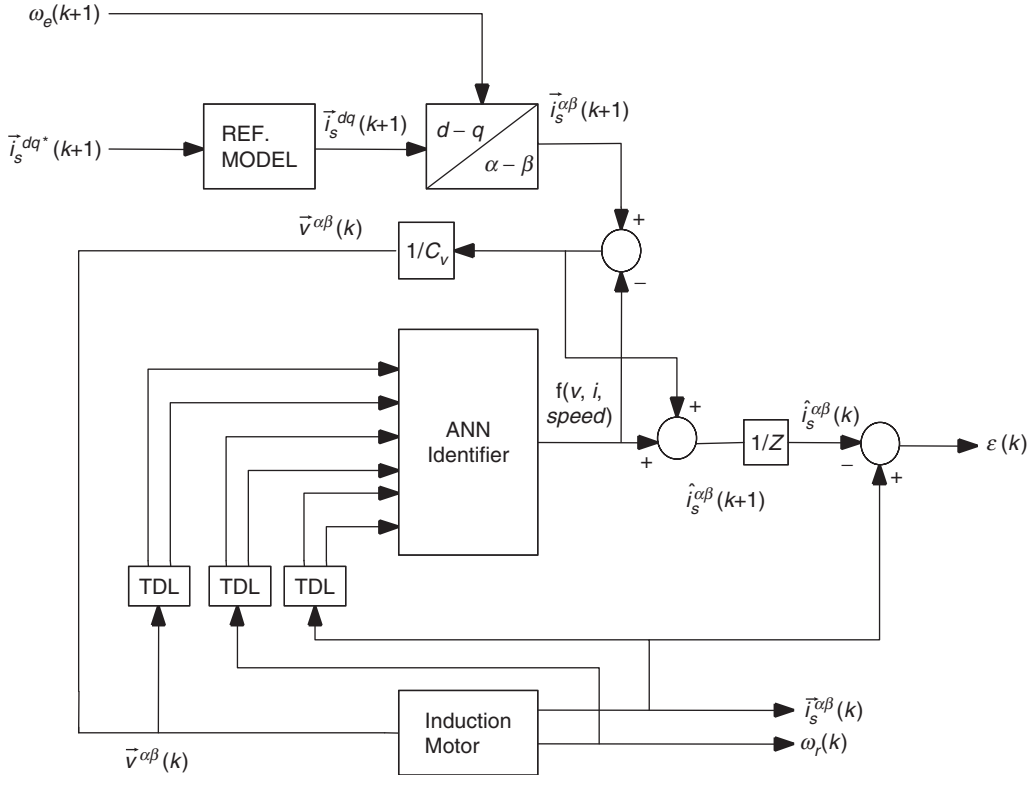

FIGURE 36.27 Adaptive current control using ANN proposed by Harley.

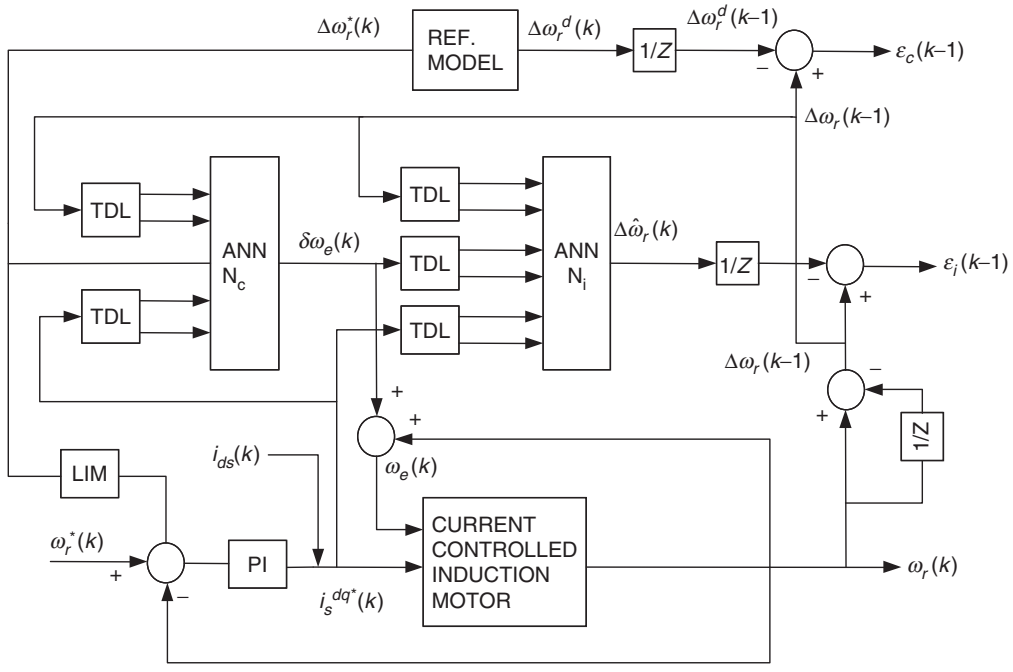

FIGURE 36.28 Adaptive speed control of the induction motor using ANN.

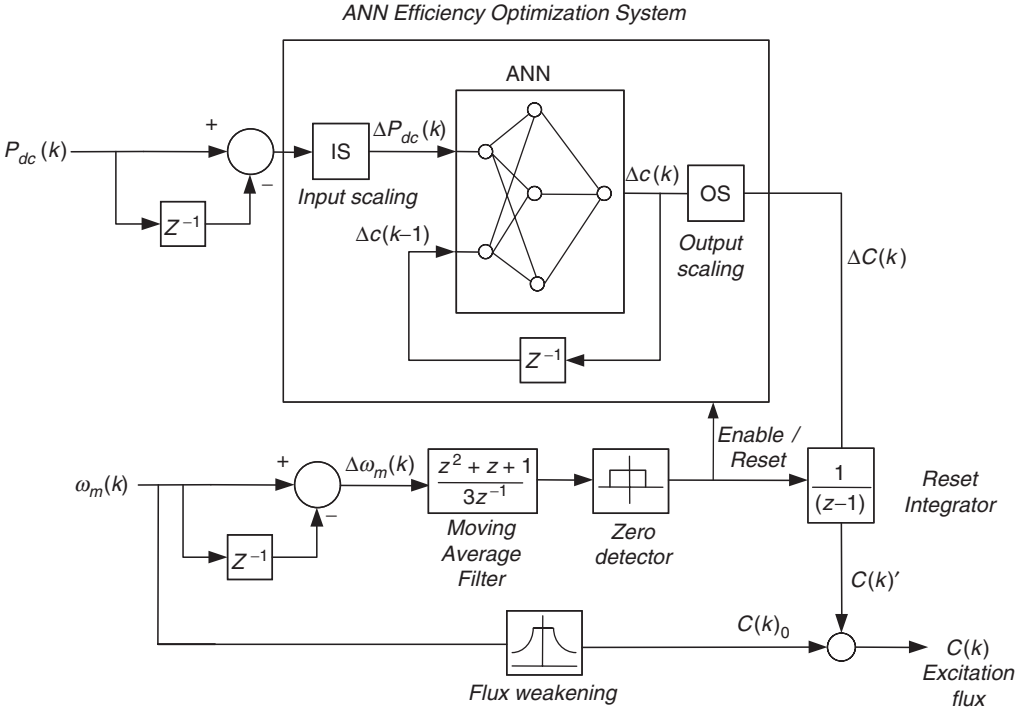

FIGURE 36.29 Induction motor efficiency optimizer using ANN.

36.4.3 Efficiency Optimization in Electric Drives

The efficiency improvement of induction motor drives via flux control can be classified into three groups: pre-computed flux programs, real-time computation of losses, and on-line input–output efficiency optimization control. All of these methods target choosing a value of the motor excitation that optimizes the motor-converter losses. The main requirement of an input–output optimization is to achieve the flux optimization in a minimum number of search steps. A constant search step may take too long if the step is too small, or it may bypass the minimum power input if the step is too large. For each mechanical operating point of the motor, it is possible to find a combination of rotor flux linkage and torque producing current i_q at which the dc link power P_{dc} to the drive system is minimum.

One of the possible ANN efficiency optimizer reported is shown in Fig. 36.29. An ANN-based search algorithm is employed to operate as an efficiency optimizer. The inputs to the system are the dc power fed into the drive system at instant k, and the change in the control variable at the previous instant $\Delta c(k-1)$. The only output is the actual change in control variable $\Delta c(k)$. Appropriate scaling from engineering units to the normalized interval $[-1, 1]$ are implemented by the input and output interfaces, input scaling (IS) and output scaling (OS). The speed signal is used to generate the reference flux $C(k)_0$ corresponding to each speed. The mechanical steady-state is detected by applying a moving average filtering to the speed variation $\Delta\omega_m$. The ANN efficiency optimizer is enabled only during the mechanical steady-state.

36.5 ANN-based Controls in Power Converters

A feedforward ANN can implement a non-linear input–output mapping. A feedforward carrier-based pulse-width modulation (PWM) technique, such as space vector modulator (SVM), can be looked at as a non-linear mapping where the command phase voltages are sampled at the input and the corresponding pulse-width patterns are established at the output. Figure 36.30 shows the block diagram of an open-loop V/f-controlled induction motor drive incorporating the proposed

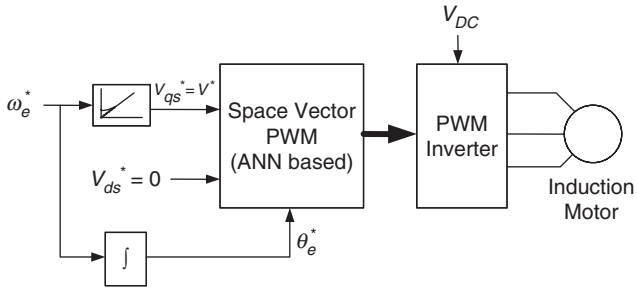

FIGURE 36.30 V/f control of induction motor using ANN-based SVM.

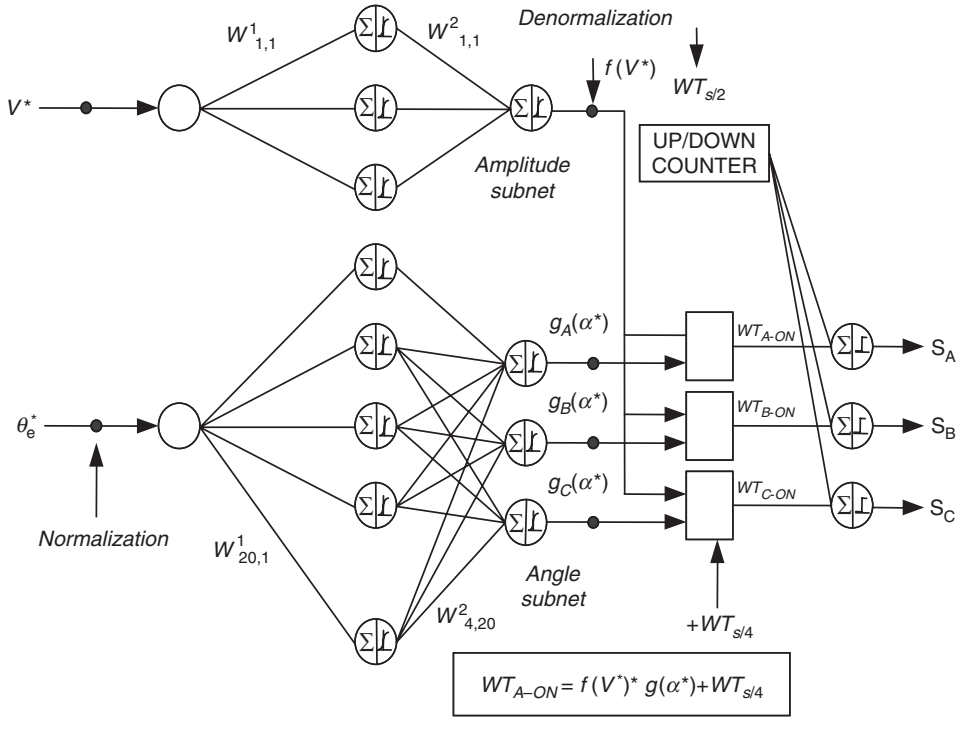

FIGURE 36.31 Neural network topology ($2 \times 20 \times 4$) for PWM wave synthesis.

ANN-based SVM controller. The command voltage $V_{qs}^{*}(=V^{*})$ is generated from the frequency or speed command, and the angle command θ_e^{*} is obtained by integrating the frequency, as shown. The output of the modulator generates the PWM patterns for the inverter switches.

The ANN can be conveniently trained off-line with the data generated by calculation of the SVM algorithm. The ANN has inherent learning capability that can give improved precision by interpolation unlike the standard lookup table method. Figure 36.31 shows such an SVM that can operate during both undermodulation and overmodulation regions linearly extending smoothly up to a square wave. The SVM is implemented using two subnets: angle subnet and amplitude subnet. The subnets use a multiplayer perceptron-type network with sigmoidal-type transfer function. The bias is not shown in the figure. The composite network uses two neurons at the input, 20 neurons in the hidden layer, and four output neurons. The input signal to the angle subnet is θ_e angle which is normalized and then pulse-width functions at unit amplitude are solved (or mapped) at the output for three phases, as indicated. The amplitude subnet implements the $f(V^{*})$ function. The digital words corresponding to the turn-on time are generated by multiplying the angle subnet output with that of the amplitude subnet and then adding the $T_s/4$ bias signal, as shown. The PWM signals are then generated using a single timer. The angle subnet is trained with an angle interval of 2.16° in the range of 0–360°. Due to learning or interpolation capability, both the subnets will operate higher signal resolution. A sampling interval T_s of 50 μs corresponds to a switching frequency of 20 kHz and a 100 μs that corresponds to 10 kHz.

Further Reading

1. P. Vas, *Artificial Intelligence-Based Electrical Machines and Drives: application of fuzzy, neural, fuzzy-neural and genetic-algorithm – based techniques*, Oxford University Press, New York, 1999.
2. B.K. Bose, *Modern Power Electronics and AC Drives*, Prentice Hall, New Jersey, 2002.
3. L. Ben-Brahim, S. Tadakuma, and A. Akdag, "Speed control of induction motor without rotational transducers," *IEEE Transactions on Industry Applications*, vol.35, no.4, pp. 844–850, July/August 1999.
4. M.G. Simoes and B.K. Bose, "Neural network based estimator of feedback signals for a vector controlled induction motor drive," *IEEE Transactions on Industry Applications*, vol.31, pp. 620–629, May/June 1995.
5. K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol.2, pp. 183–192, 1989.
6. D.T. Pham and X. Liu, *Neural Networks for Identification, Prediction and Control*, Springer–Verlag, New York, 1995.
7. M. Wishart and R.G. Harley, "Identification and control of induction machines using artificial neural networks," *IEEE Transaction on Industry Applications*, vol.31, no.3, pp. 612–619, May/June 1995.
8. K.S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol.1, no.1, pp. 4–27, March 1990.
9. J.O. Pinto, B.K. Bose, L.E.B. De Silva, and M.P. Kazmierkowski, "A Neural-Network based Space-Vector PWM Controller for Voltage-fed Inverter induction motor drive," *IEEE Transactions on Industry Applications*, vol.36, no.6, pp. 1628–1636, November/December 2000.