International Journal of Innovative
Computing, Information and Control
Volume **11**, Number **2**, April **2015**

ICIC International ©2015 ISSN 1349-4198
pp. 433–446

FUZZY NEURAL NETWORK SPEED ESTIMATION METHOD FOR INDUCTION MOTOR SPEED SENSORLESS CONTROL

Chun-Hao Lu[1], Cheng-Chi Tai[1], Tien-Chi Chen[2] and Wei-Chung Wang[2]

[1]Department of Electrical Engineering
National Cheng Kung University
No. 1, University Road, Tainan 701, Taiwan

[2]Department of Computer and Communication
Kun Shan University
No. 195, Kunda Road, Yongkang Dist., Tainan 71003, Taiwan
tchichen@mail.ncku.edu.tw

Received March 2014; revised July 2014

Abstract. *Field-oriented induction motor control presents high static, dynamic performance. Precise rotor speed information is critical for induction motor control to achieve speed loop feedback control. In the past encoders were widely used to obtain the speed information for the induction motor. However, a speed sensor would increase the cost of the entire system and reduce system reliability. In addition, for some special applications (such as very high speed motor drives) difficulties were encountered in mounting these speed sensors. Sensorless speed control would overcome these problems. This paper proposes a fuzzy neural network speed estimation method for induction motor speed sensorless control. The speed estimation is based on rotor flux deduction and estimated rotor flux, calculated using a fuzzy neural network. The fuzzy neural network is a four-layer network. The steepest descent algorithm is used to adjust the fuzzy neural network parameters to minimize the error between the rotor flux and estimated rotor flux, enabling precise rotor speed estimation.*

Keywords: Induction motor, Encoder, Speed sensorless control, Fuzzy neural network speed estimation, Steepest descent algorithm

1. Introduction.

Servo systems have become indispensable to many industry factory automation applications. The DC motor and induction motor were previously applied in many industries. The DC motor has been used extensively for variable speed control, e.g., industrial robots and numerically controlled machinery, because of their simple modeling and ease of control [1,2]. However, DC motors have big volume, complicated structure and certain commutator and brush disadvantages. The induction motor does not need brushes and commutators and does not have the drawbacks mentioned above with the advantages of a simple, rugged structure, reliability and easy maintenance [3]. Therefore, the induction motor has replaced DC motors in many industrial applications [4,5]. Because of the advances in power electronics and microprocessors, induction motor applications in speed control have become more attractive.

A control scheme is important for precise induction motor control. The V/f control method was used in induction motor speed control [6,7]. However, due to stator resistance and necessary rotor slip influence to produce torque, its application at low speeds is still challenging. To overcome the difficulty mentioned above a closed-loop control scheme for the induction motor was presented. To solve the foregoing problems, induction motors have been controlled like DC motors using a field-oriented control (FOC) approach, which has high static, dynamic performance [8-10]. FOC is based on decoupling torque and rotor flux through nonlinear coordinate transformation. Therefore, the rotor speed is linearly related to the torque current after the rotor flux attains steady-state values. The FOC applied to induction motor drives allows us to perform fast and fully decoupled torque and flux control. FOC algorithms need to know the rotor flux angular position to correctly align the stator current vector to obtain decoupled control. As a consequence, it is possible to control the torque and rotor flux as in DC motor control by acting on two separate stator current components [11].

In modern induction motor drive control techniques the closed-loop speed control system uses a shaft encoder to measure the motor's speed. However, a speed sensor has several disadvantages from the drive cost, noise immunity and reliability viewpoints. In addition, for some special applications such as very high speed motor drives, some difficulties are encountered in mounting these speed sensors. Sensorless speed control is preferable from the low cost drive point of view [12,13]. Numerous researches have recently been published on sensorless induction motor speed control [14-16]. These methods are further classified into the following methodologies: extended Kalman filter techniques [17], model reference adaptive systems [18] and sliding mode method [19]. In the extended Kalman filter technique the composite states consist of the rotor fluxes and rotor speed. The extended Kalman filter is employed to identify induction motor speed based on measured quantities such as the stator currents. However, the Kalman filtering algorithm does not contain a feedback signal to train the parameter that would increase the system uncertainty. The reference model and adjustable model are interchangeable for the model reference systems, which can be used to identify the rotor speed of an induction machine. However, the model reference adaptive system sensorless speed methods are affected mainly by the motor's parameters which affect the speed estimation accuracy, which could spoil the system stability. In the sliding mode method a direct torque and flux control strategy based on a sliding-mode observer using a dual reference frame motor model is introduced. However, in the sliding mode method the motor's parameters will affect the speed estimation accuracy.

Fuzzy neural network control has recently become an active research area. Because of the adaptive abilities in a network learning process, applying neural networks to control systems have become a promising alternative to process control. The fuzzy neural network control can be applied to the AC servo systems since they approximate any desired degree of accuracy with a wide range of nonlinear models. The fuzzy neural networks can be classified as feed-forward fuzzy neural network and recurrent fuzzy neural network. A feed-forward fuzzy neural network can approximate any continuous functions closely. The feed-forward fuzzy neural network is a static mapping. Moreover, the internal information in a feed-forward fuzzy neural network can be utilized for weight updates and the function approximation is sensitive to the training data. Feed-forward fuzzy neural network control methods were developed for induction motor speed control to provide good speed response when the motor is operated under diverse operating conditions and parameter variations. Up to now the fuzzy neural network has been applied mainly in control systems, such as motor drive systems, time delay systems, robot manipulators and chaotic systems [20-26]. Its application in speed estimation is practically new [27].

This paper proposes a speed estimation algorithm based on the fuzzy neural network. The conventional fuzzy neural networks were the feed forward multilayer type where no information is fed back during application. However, the feedback signal is necessary during the training process. Unlike the conventional fuzzy neural network the proposed fuzzy neural network has a feedback signal composed of a four-layer network including an input layer, membership layer, rule layer and output layer. The rotor flux is derived from the motor's dynamic model. The estimated rotor flux is the fuzzy neural network output.

The error between the rotor flux and the estimated rotor flux is used as the feedback signal to adjust the fuzzy neural network parameters through a back-propagated method [28]. This method minimizes the difference between the rotor flux and the estimated rotor flux. The back-propagation mechanism is easily derived so that precise rotor speed estimation can quickly track the actual motor speed.

Microprocessor techniques have become more mature in recent years. The FOC and feedback signal processes are undoubtedly complex, requiring a powerful microprocessor or digital-signal-processor (DSP). The proposed control scheme is implemented in TMS320F2808 DSP. Experimental results are shown to confirm that the proposed fuzzy neural network speed estimation can provide good performance over a wide speed range for induction motor speed control. The proposed algorithm uses the estimated speed to quickly track the actual speed. The proposed sensorless speed control scheme with and without load presents good performance.

2. **The Dynamic Model of Induction Motor.**

The induction motor dynamic model in the synchronous rotating *d-q* frame can be expressed as follows [29]:

$$\frac{d}{dt}\begin{bmatrix} i_{ds}^e \\ i_{qs}^e \\ \phi_{dr}^e \\ \phi_{qr}^e \end{bmatrix} = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s}+\frac{R_r(1-\sigma)}{\sigma L_r}\right) & \omega_e & \frac{L_m R_r}{\sigma L_s L_r^2} & \frac{\omega_r L_m}{\sigma L_s L_r} \\ -\omega_e & -\left(\frac{R_s}{\sigma L_s}+\frac{R_r(1-\sigma)}{\sigma L_r}\right) & -\frac{\omega_r L_m}{\sigma L_s L_r} & \frac{L_m R_r}{\sigma L_s L_r^2} \\ \frac{L_m R_r}{L_r} & 0 & -\frac{R_r}{L_r} & (\omega_e-\omega_r) \\ 0 & \frac{L_m R_r}{L_r} & -(\omega_e-\omega_r) & -\frac{R_r}{L_r} \end{bmatrix} \begin{bmatrix} i_{ds}^e \\ i_{qs}^e \\ \phi_{dr}^e \\ \phi_{qr}^e \end{bmatrix} + \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_{ds}^e \\ v_{qs}^e \end{bmatrix} \tag{1}$$

The torque equation is given as follows:

$$T_e = \frac{3PL_m}{4L_r}(\phi_{dr}^e i_{qs}^e - \phi_{qr}^e i_{ds}^e) = \frac{2}{P}J\frac{d\omega_r}{dt} + B\omega_r + T_L \tag{2}$$

where L_s, L_r, L_m: stator inductance, rotor inductance and mutual inductance, R_s, R_r: stator resistance and rotor resistance, $\sigma = 1-(L_m^2/L_sL_r)$, v_{qs}^e, v_{ds}^e: q-axis and d-axis stator voltage in the synchronous rotating frame, i_{qs}^e, i_{ds}^e: q-axis and d-axis stator current in the synchronous rotating frame, ϕ_{qr}^e, ϕ_{dr}^e: q-axis and d-axis rotor flux in the synchronous rotating frame, P: pole number of the induction motor, T_e, T_L: electromagnetic torque and load torque, J, B: inertial and viscous moment coefficient of the induction motor, ω_r: rotor angular velocity, ω_e: electrical angular velocity.

Manipulating the first and third row of (1) yields:

$$\frac{d}{dt}\phi_{dr}^e = \frac{L_r}{L_m}\left(v_{ds}^e - R_s i_{ds}^e - \sigma L_s \frac{d}{dt} i_{ds}^e\right) + \frac{\sigma L_r L_s \omega_e}{L_m} i_{qs}^e + \omega_e \phi_{qr}^e \tag{3}$$

Performing algebraic operations on the second and forth rows of Equation (1) yields:

$$\frac{d}{dt}\phi_{qr}^e = \frac{L_r}{L_m}\left(v_{qs}^e - R_s i_{qs}^e - \sigma L_s \frac{d}{dt} i_{qs}^e\right) - \frac{\sigma L_s L_r \omega_e}{L_m} i_{ds}^e - \omega_e \phi_{dr}^e \tag{4}$$

Equations (3) and (4) can be rewritten as the following matrix form of the rotor flux equation:

$$\frac{d}{dt}\boldsymbol{\varphi}_r^e = \frac{L_r}{L_m} I \left(\boldsymbol{V}_s^e - R_s \boldsymbol{I}_s^e - \sigma L_s \frac{d}{dt}\boldsymbol{I}_s^e\right) + \frac{\sigma L_r L_s \omega_e}{L_m}\boldsymbol{J}\boldsymbol{I}_s^e + \omega_e \boldsymbol{J}\boldsymbol{\varphi}_r^e \tag{5}$$

where $\boldsymbol{\varphi}_r^e = \left[\phi_{dr}^e, \phi_{qr}^e\right]^T$, $\boldsymbol{V}_s^e = \left[v_{ds}^e, v_{qs}^e\right]^T$, $\boldsymbol{I}_s^e = \left[i_{ds}^e, i_{qs}^e\right]^T$, $\boldsymbol{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\boldsymbol{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

3. **Fuzzy Neural Network Speed Estimation.** A fuzzy neural network is employed for induction motor speed estimation. Figure 1 illustrates a block diagram of the proposed fuzzy neural network sensorless speed estimation method. Two independent fluxes are used in the proposed method. The first is the dynamic model rotor flux ($\boldsymbol{\varphi}_r^e$). The second is the estimated rotor flux ($\widehat{\varphi}_r^e$) obtained from the fuzzy neural network. The error ($\vec{e}$) between the two independent fluxes is used to adjust the fuzzy neural network parameters ($\bar{y}^j, \bar{x}_i^j, \bar{\sigma}_i^j$) using the steepest descent algorithm such that the estimated rotor flux coincides with the rotor flux. The estimated speed ($\widehat{\omega}_r$) can precisely track the actual motor speed (ω_r).

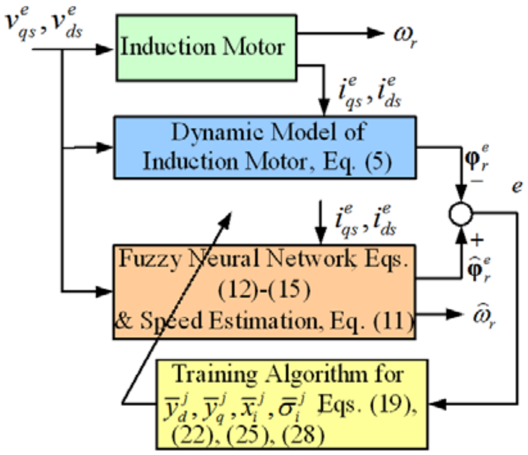

v^e_{qs}, v^e_{ds}
Induction Motor
ω_r
i^e_{qs}, i^e_{ds}
Dynamic Model of Induction Motor, Eq. (5)
φ^e_r
−
e
i^e_{qs}, i^e_{ds}
+
$\hat{\varphi}^e_r$
Fuzzy Neural Network Eqs. (12)-(15) & Speed Estimation, Eq. (11)
$\hat{\omega}_r$
Training Algorithm for $\bar{y}^j_d, \bar{y}^j_q, \bar{x}^j_i, \bar{\sigma}^j_i$ Eqs. (19), (22), (25), (28)

FIGURE 1. A fuzzy neural network for induction motor speed estimation

3.1. **The speed estimation principle.** The estimated rotor flux equation is derived form the third and the forth row of Equation (1). Taking some algebraic operations from the third and fourth rows of Equation (1), the estimated rotor flux equation can be expressed in the following matrix form:

$$\frac{d}{dt}\widehat{\boldsymbol{\varphi}}_r^e = \left[\frac{-1}{\tau_r}\boldsymbol{I} + (\widehat{\omega}_r - \omega_e)\boldsymbol{J}\right]\widehat{\boldsymbol{\varphi}}_r^e + \frac{L_m}{\tau_r}\boldsymbol{I}\,\boldsymbol{I}_s^e \tag{6}$$

where $\widehat{\boldsymbol{\varphi}}_r^e = \left[\widehat{\phi}_{dr}^e,\ \widehat{\phi}_{qr}^e\right]^T$, $\tau_r = L_r/R_r$ is the rotor time constant, and $\widehat{\omega}_r$ is the estimated rotor speed.

By applying the backward difference method, Equation (6) can be discretized and described as follows:

$$\frac{1-z^{-1}}{T}\widehat{\boldsymbol{\varphi}}_r^e(z) = \left[\frac{-1}{\tau_r}\boldsymbol{I} + (\widehat{\omega}_r(z) - \omega_e(z))\boldsymbol{J}\right]\widehat{\boldsymbol{\varphi}}_r^e(z) + \frac{L_m}{\tau_r}\boldsymbol{I}\,\boldsymbol{I}_s^e(z) \tag{7}$$

where T is the sampling time.

Taking the inverse z-transform of (7), the discrete-time form of Equation (6) can be expressed as:

$$\begin{aligned}\widehat{\boldsymbol{\varphi}}_r^e(k) = &\left(1 - \frac{T}{\tau_r}\right)\boldsymbol{I}\widehat{\boldsymbol{\varphi}}_r^e(k-1) - \omega_e(k-1)T\boldsymbol{J}\widehat{\boldsymbol{\varphi}}_r^e(k-1) \\ &+ \hat{\omega}_r(k-1)T\boldsymbol{J}\widehat{\boldsymbol{\varphi}}_r^e(k-1) + \frac{L_mT}{\tau_r}\boldsymbol{I}\,\boldsymbol{I}_s^e(k-1)\end{aligned} \tag{8}$$

Since the estimated rotor speed $\widehat{\omega}_r$ is unknown and may vary with time, the estimation process becomes time varying due to the unknown term $(\widehat{\omega}_r(k)TJ\widehat{\phi}^e_r(k))$ in Equation (8). Forming the proposed fuzzy neural network estimator as a four-layer structure resolves the problem.

The third term of Equation (8) is expressed as:

$$\widehat{\boldsymbol{y}}_f(k) = \widehat{\omega}_r(k)T\boldsymbol{J}\widehat{\boldsymbol{\varphi}}^e_r(k) \tag{9}$$

where $\widehat{\boldsymbol{y}}_f(k) = \left[\ \widehat{y}_{fd}(k) \quad \widehat{y}_{fq}(k)\ \right]^T$. By multiplying $\widehat{\boldsymbol{\varphi}}^{eT}_r$ on both sides of Equation (9), the solution can be expressed as:

$$\widehat{\boldsymbol{\varphi}}^{eT}_r\ \widehat{y}_f(k) = \widehat{\boldsymbol{\varphi}}^{eT}_r\ \widehat{\omega}_r(k)T\boldsymbol{J}\widehat{\boldsymbol{\varphi}}^e_r(k) \tag{10}$$

Any mismatch between the rotor flux $\boldsymbol{\varphi}^e_r(k)$ and the estimated flux $\widehat{\boldsymbol{\varphi}}^e_r(k)$ estimated by the fuzzy neural network system would automatically produce an error. This error is further used to adjust the fuzzy neural network parameters. If $\boldsymbol{\varphi}^e_r(k)$ is equal to $\widehat{\boldsymbol{\varphi}}^e_r(k)$, the estimated rotor speed $\widehat{\omega}_r$ can be obtained as:

$$\widehat{\omega}_r(k) = \frac{\widehat{\boldsymbol{\varphi}}^{eT}_r(k)\widehat{\boldsymbol{y}}_{\boldsymbol{f}}(k-1)}{T\left(\widehat{\phi}^{e2}_{qr}(k) + \widehat{\phi}^{e2}_{dr}(k)\right)} \tag{11}$$

In this way, the motor speed can be predicted accurately by the fuzzy neural network sensorless speed estimation.

3.2. **Fuzzy neural network structure.** A four-layer fuzzy neural network, as shown in Figure 2, including an input layer, a membership layer, a rule layer and an output layer, is used to implement the fuzzy neural network. The fuzzy neural network input is $x_1(k) = v^e_{ds}(k)$, $x_2(k) = v^e_{qs}(k)$, $x_3(k) = i^e_{ds}(k)$, $x_4(k) = i^e_{qs}(k)$. The output of every node in the input layer is equal to the input. Each node in the membership layer performs a membership function. The Gaussian function is selected as the membership function, it

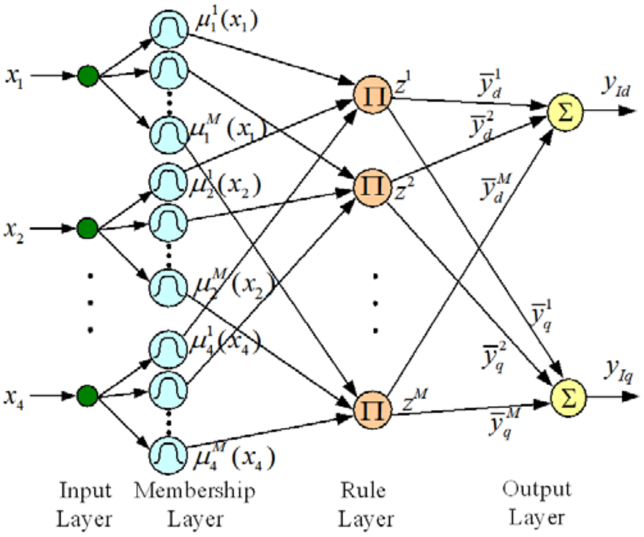


Figure 2. The four-layer fuzzy neural network structure

can be described as:

$$\mu_i^j(x_i(k)) = \exp\left[-\left(\frac{x_i(k)-\bar{x}_i^j(k)}{\bar{\sigma}_i^j(k)}\right)^2\right] \tag{12}$$

where $j = 1, \ldots, M$, and M are the number of membership functions in each input node. The value of M is set to 4, $\bar{x}_i^j(k)$ and $\bar{\sigma}_i^j(k)$ are, respectively, the mean and standard deviation of the Gaussian function.

Each node in the rule layer is denoted by Π, which multiplies all input signals. The rule layer output for the j node is expressed as follows:

$$z^j(k) = \prod_{i=1}^{4} \exp\left[-\left(\frac{x_i(k)-\bar{x}_i^j(k)}{\sigma_i^j(k)}\right)^2\right] \tag{13}$$

The signal node in the output layer is labeled as $\sum$, which computes the summation of all input signals and the output layer output is expressed as follows:

$$y_{Id}(k) = \sum_{j=1}^{M} \bar{y}_d^j(k) z^j(k) \tag{14}$$

$$y_{Iq}(k) = \sum_{j=1}^{M} \bar{y}_q^j(k) z^j(k) \tag{15}$$

where $y_{Id}(k) = \overset{\frown}{\phi}{}_{dr}^{e}(k)$ and $y_{Iq}(k) = \overset{\frown}{\phi}{}_{qr}^{e}(k)$.

3.3. **Training algorithm for fuzzy neural network.** This section describes the fuzzy neural network online training algorithm using the back-propagation training algorithm. The error function is defined as

$$E_I(k) = \frac{1}{2}\left(\boldsymbol{y}_I(k) - \boldsymbol{y}(k)\right)^T \left(\boldsymbol{y}_I(k) - \boldsymbol{y}(k)\right) \tag{16}$$

where $\boldsymbol{y}_I(k) = [y_{Id}(k),\ y_{Iq}(k)]^T = \widehat{\boldsymbol{\varphi}}_r^e(k) = \left[\overset{\frown}{\phi}{}_{dr}^{e}(k),\ \overset{\frown}{\phi}{}_{qr}^{e}(k)\right]^T$ is the fuzzy neural network output and $\boldsymbol{y} = [y_d(k),\ y_q(k)]^T = \boldsymbol{\varphi}_r^e(k) = \left[\phi_{dr}^e(k),\ \phi_{qr}^e(k)\right]^T$.

The objective is to train the fuzzy neural network such that $E_I(k)$ is minimized. Hence, the identification problem now becomes to train the parameters $\bar{y}_d^j(k)$, $\bar{y}_q^j(k)$, $\bar{x}_i^j(k)$ and $\bar{\sigma}_i^j(k)$ of the fuzzy neural network.

The training method is based on the steepest descent algorithm. The training algorithm derivation is described as follows.

(a) Training algorithm for $\bar{y}_d^j(k)$:

In order to train $\bar{y}_d^j(k)$, the steepest descent algorithm is expressed as follows:

$$\bar{y}_d^j(k+1) = \bar{y}_d^j(k) - \alpha_I \left.\frac{\partial E_I}{\partial \bar{y}_d^j}\right|_k \tag{17}$$

where α_I is the fuzzy identifier learning rate.

Using the chain rule, Equation (17) can be expressed as:

$$\left.\frac{\partial E_I}{\partial \bar{y}_d^j}\right|_k = \left.\left(\frac{\partial E_I}{\partial y_{Id}} \frac{\partial y_{Id}}{\partial \bar{y}_d^j}\right)\right|_k \tag{18}$$

Substituting Equations (14) and (16) into (18), then combining Equation (17) and readjusting it, the training algorithm for $\bar{y}_d^j$ can be expressed as:

$$\bar{y}_d^j(k+1) = \bar{y}_d^j(k) - \alpha_I \left[y_{Id}(k) - y_d(k)\right] z^j(k) \tag{19}$$

(b) Training algorithm for $\bar{y}_q^j(k)$:

In order to train $\bar{y}_q^j(k)$, the steepest descent algorithm is expressed as follows:

$$\bar{y}_q^j(k+1) = \bar{y}_q^j(k) - \alpha_I \left.\frac{\partial E_I}{\partial \bar{y}_q^j}\right|_k \tag{20}$$

Using the chain rule, Equation (20) can be expressed as:

$$\left.\frac{\partial E_I}{\partial \bar{y}_q^j}\right|_k = \left.\left(\frac{\partial E_I}{\partial y_{Iq}}\frac{\partial y_{Iq}}{\partial \bar{y}_q^j}\right)\right|_k \tag{21}$$

Substituting (15) and (16) into (21), then combining Equation (20) and readjusting it, the training algorithm for $\bar{y}_q^j(k)$ can be expressed as:

$$\bar{y}_q^j(k+1) = \bar{y}_q^j(k) - \alpha_I \left[y_{Iq}(k) - y_q(k)\right] z^j(k) \tag{22}$$

(c) Training algorithm for $\bar{x}_i^j(k)$:

For training $\bar{x}_i^j(k)$, the steepest descent algorithm can be expressed as follows:

$$\bar{x}_i^j(k+1) = \bar{x}_i^j(k) - \alpha_I \left.\frac{\partial E_I}{\partial \bar{x}_i^j}\right|_k \tag{23}$$

Using the chain rule, Equation (23) can be expressed as:

$$\left.\frac{\partial E_I}{\partial \bar{x}_i^j}\right|_k = \left.\left[\left(\frac{\partial E_I}{\partial y_{Id}}\frac{\partial y_{Id}}{\partial z^j} + \frac{\partial E_I}{\partial y_{Iq}}\frac{\partial y_{Iq}}{\partial z^j}\right)\frac{\partial z^j}{\partial \bar{x}_i^j}\right]\right|_k \tag{24}$$

Substituting (13), (14), (15) and (16) into (24), then combining Equation (23) and readjusting it, the training algorithm for $\bar{x}_i^j$ can be expressed as:

$$\bar{x}_i^j(k+1) = \bar{x}_i^j(k) - \alpha_I \frac{2\left[\left(y_{Id}(k) - y_d(k)\right)\bar{y}_d^j + \left(y_{Iq}(k) - y_q(k)\right)\bar{y}_q^j(k)\right] z^j(k)\left[x_i(k) - \bar{x}_i^j(k)\right]}{\left(\sigma_i^j(k)\right)^2} \tag{25}$$

(d) Training algorithm for $\bar{\sigma}_i^j(k)$:

For training $\bar{\sigma}_i^j(k)$, the steepest descent algorithm can be expressed as follows:

$$\bar{\sigma}_i^j(k+1) = \bar{\sigma}_i^j(k) - \alpha_I \left.\frac{\partial E_I}{\partial \bar{\sigma}_i^j}\right|_k \tag{26}$$

Using the chain rule, Equation (26) can be expressed as:

$$\left.\frac{\partial E_I}{\partial \bar{\sigma}_i^j}\right|_k = \left.\left[\left(\frac{\partial E_I}{\partial y_{Id}}\frac{\partial y_{Id}}{\partial z^j} + \frac{\partial E_I}{\partial y_{Iq}}\frac{\partial y_{Iq}}{\partial z^j}\right)\frac{\partial z^j}{\partial \bar{\sigma}_i^j}\right]\right|_k \tag{27}$$

Substituting (13), (14), (15) and (16) into (27), then combining Equation (26) and readjusting it, the training algorithm for $\bar{\sigma}_i^j(k)$ can be expressed as:

$$\bar{\sigma}_i^j(k+1) = \bar{\sigma}_i^j(k) - \alpha_I \frac{2\left[\left(y_{Id}(k) - y_d(k)\right)\bar{y}_d^j + \left(y_{Iq}(k) - y_q(k)\right)\bar{y}_q^j(k)\right] z^j(k)\left[x_i(k) - \bar{x}_i^j(k)\right]^2}{\left(\bar{\sigma}_i^j(k)\right)^3} \tag{28}$$

The training algorithms given in (19), (22), (25) and (28) perform a steepest descent algorithm for the fuzzy neural network.

4. **Experiments.** Experiments were necessary to demonstrate the feasibility of the proposed control scheme. The block diagram of the indirect FOC method and fuzzy neural network speed estimation method for induction motor speed sensorless control is shown in Figure 3. The experimental software control program includes the field-oriented control, coordinate translator, adaptive current control PWM, fuzzy neural network speed estimation method. The induction motor parameters are: $R_s = 1.1\Omega$, $R_r = 1.3\Omega$, $L_s = 0.1452$H, $L_r = 0.1456$H, $L_m = 0.1363$H, $J = 6.8 \times 10^{-4}\text{kg} \cdot \text{m}^2$, $B = 5.15 \times 10^{-4}\text{N} \cdot \text{m} \cdot \text{s/rad}$, $P = 2$.

A block diagram of the experiment configuration is shown in Figure 4. The experimental equipment includes the induction motor driver: converter and inverter, isolated circuit, Hall current sensor circuit and TMS320F2808 DSP. An indirect field-oriented control method is used for induction motor speed control. The proposed fuzzy neural network speed estimation method for induction motor speed sensorless control scheme and indirect field-oriented control method are implemented in TMS320F2808 DSP.

a. Speed tracking control at 200 rpm without loading

For a lower speed command of 200 rpm with no load, the actual speed and estimated speed responses are shown in Figures 5(a) and 5(b). The speed error between the actual speed and estimated speed is shown in Figure 5(c). According to Figures 5(a)-5(c), the actual motor speed has good transient response and the estimated motor speed can quickly track the actual motor speed. The speed error between the actual speed and estimated speed decays very soon in transient state and is very slight in steady state.

The actual d-axis rotor flux and estimated d-axis rotor flux responses are shown in Figures 5(d) and 5(e). The d-axis rotor flux error between the actual d-axis rotor flux and estimated d-axis rotor flux are shown in Figure 5(f). The actual q-axis rotor flux and estimated q-axis rotor flux responses are shown in Figures 5(g) and 5(h). The q-axis rotor

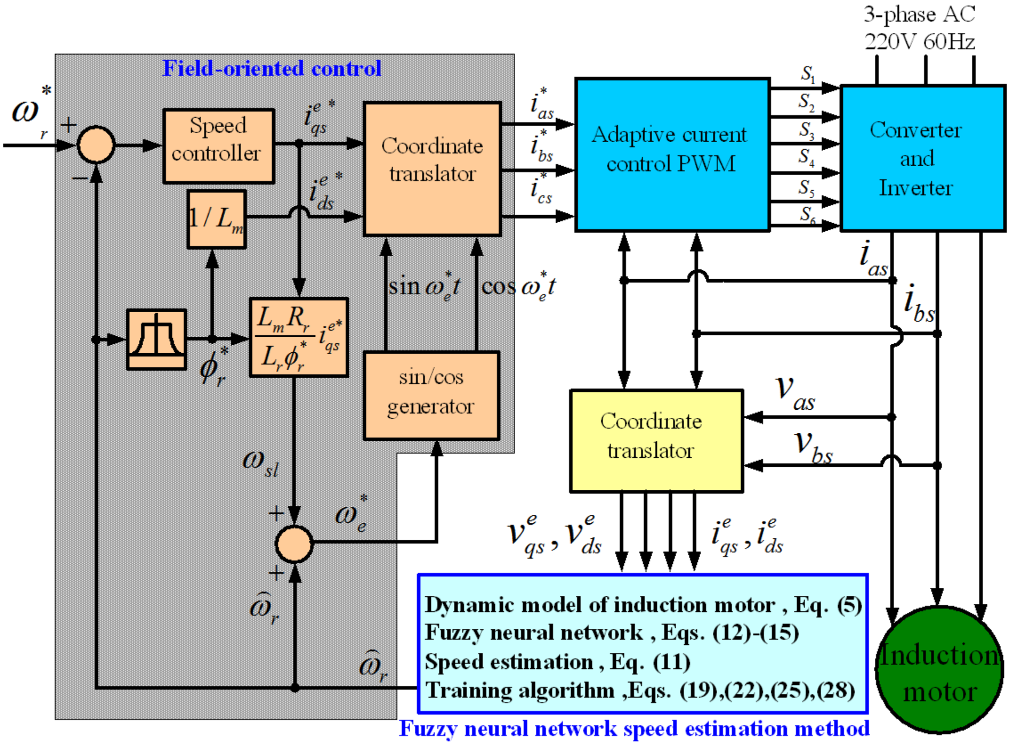


Figure 3. The block diagram of the indirect FOC method and fuzzy neural network speed estimation method for induction motor speed sensorless control

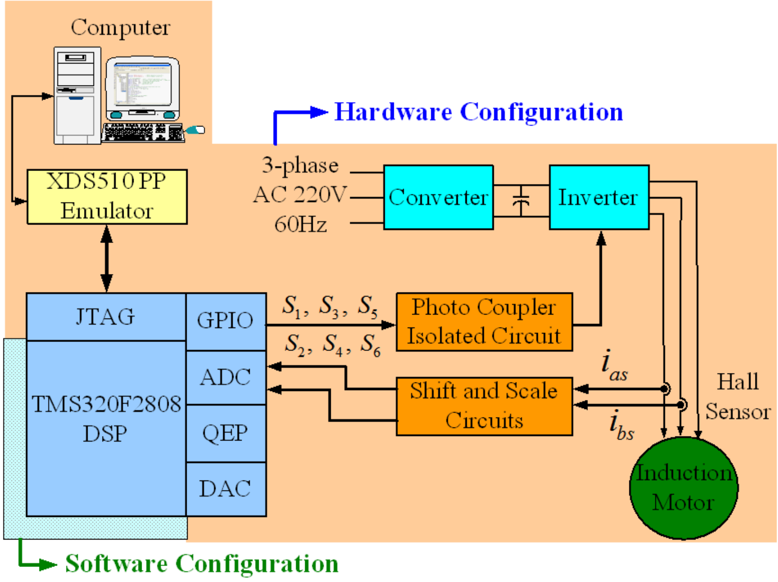


FIGURE 4. Block diagram of the experiment configuration

flux error between the actual q-axis rotor flux and estimated q-axis rotor flux is shown in Figure 5(i). Figure 5(j) reveals the rotor flux locus in d-q frame. According to Figures 5(d) to 5(j), the estimated d-axis and q-axis rotor fluxes can quickly track the actual d-axis and q-axis rotor fluxes, respectively. The d-axis rotor flux error and q-axis rotor flux error are very slight in steady state. The rotor flux locus in d-q frame indicate that the rotor rotates smoothly. The experimental results reveal that the training algorithms given in (19), (22), (25) and (28) perform a steepest descent algorithm for the fuzzy neural network such that $E_I(k)$ is minimized.

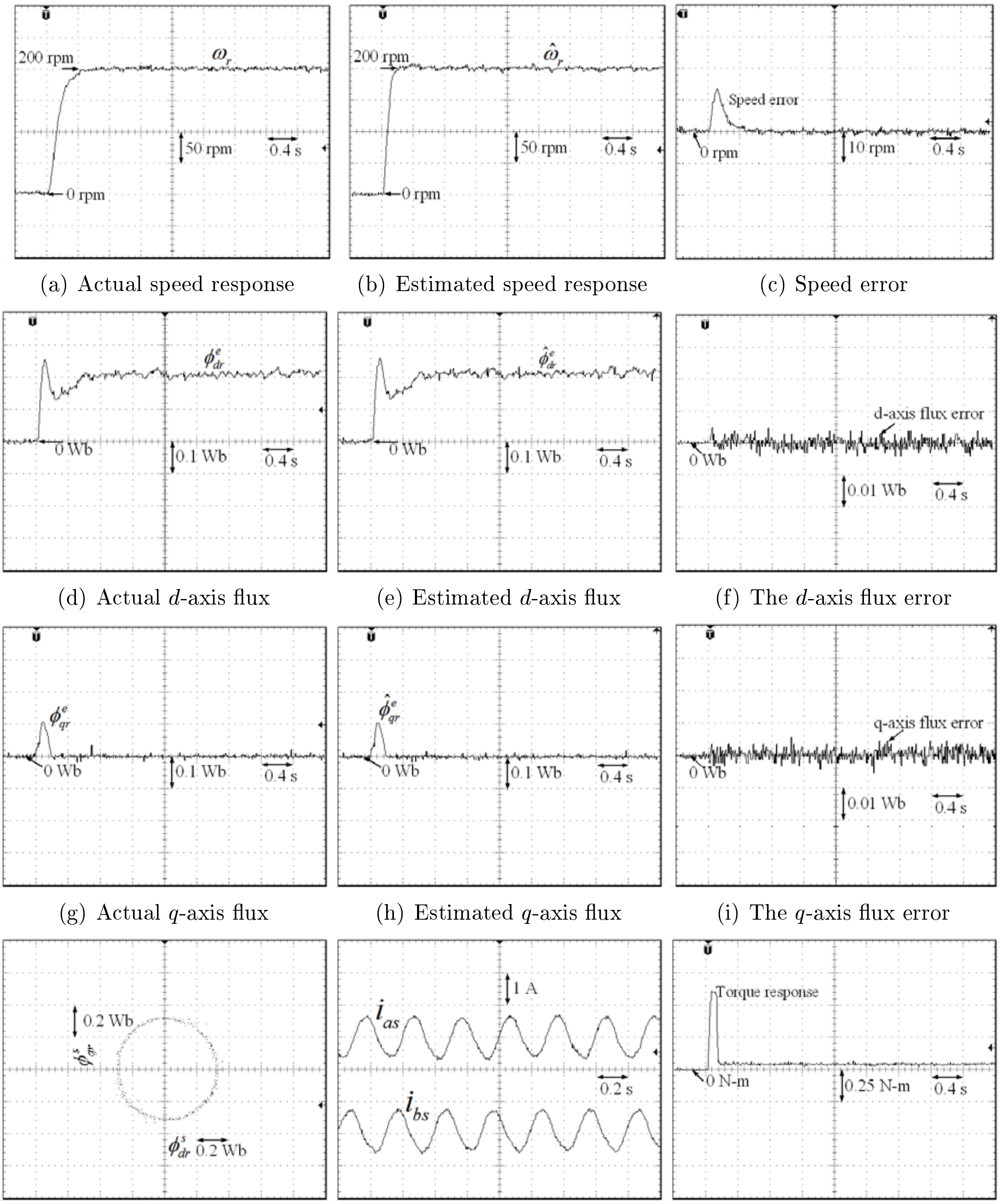
Figure 5(k) shows the phase-a and phase-b stator currents. According to Figure 5(k), the current waves are very similar to the sinusoidal signal. In other words, the current signals do not contain much harmonic frequency. Figure 5(l) shows the torque response. The experimental results show that the proposed algorithm has fairly good performance.

b. Speed tracking control at 1000 rpm without loading

For a higher speed command of 1000 rpm and with no load, the actual speed response, estimated speed response and speed error between the actual speed and estimated speed are shown in Figures 6(a)-6(c). As one can see in Figures 6(a) to 6(c), the actual motor speed has a fairly transient response and the estimated motor speed can quickly track the actual motor speed. The speed error decays very soon and is very slight in the steady state. Both the actual and estimated speeds have great steady state response.

The actual d-axis rotor flux and estimated d-axis rotor flux are shown in Figures 6(d) and 6(e). The d-axis rotor flux error between the actual d-axis rotor flux and estimated d-axis rotor flux is shown in Figure 6(f). The actual q-axis rotor flux and estimated q-axis rotor flux are shown in Figures 6(g) and 6(h). The q-axis rotor flux error between the actual q-axis rotor flux and estimated q-axis rotor flux is shown in Figure 6(i). Figure 6(j) reveals the rotor flux locus in d-q frame. These figures show that the estimated d-axis and q-axis rotor fluxes can quickly track the actual d-axis and q-axis rotor fluxes, which indicates that the training algorithms given in (19), (22), (25) and (28) perform a steepest descent algorithm for the fuzzy neural network such that $E_I(k)$ is minimized.

Figure 6(k) shows the phase-a and phase-b stator currents. Figure 6(l) shows the torque response. The current waves are very similar to the sinusoidal signal, which do not contain much harmonic frequency. This experiment shows that the proposed algorithm has fairly



(a) Actual speed response (b) Estimated speed response (c) Speed error

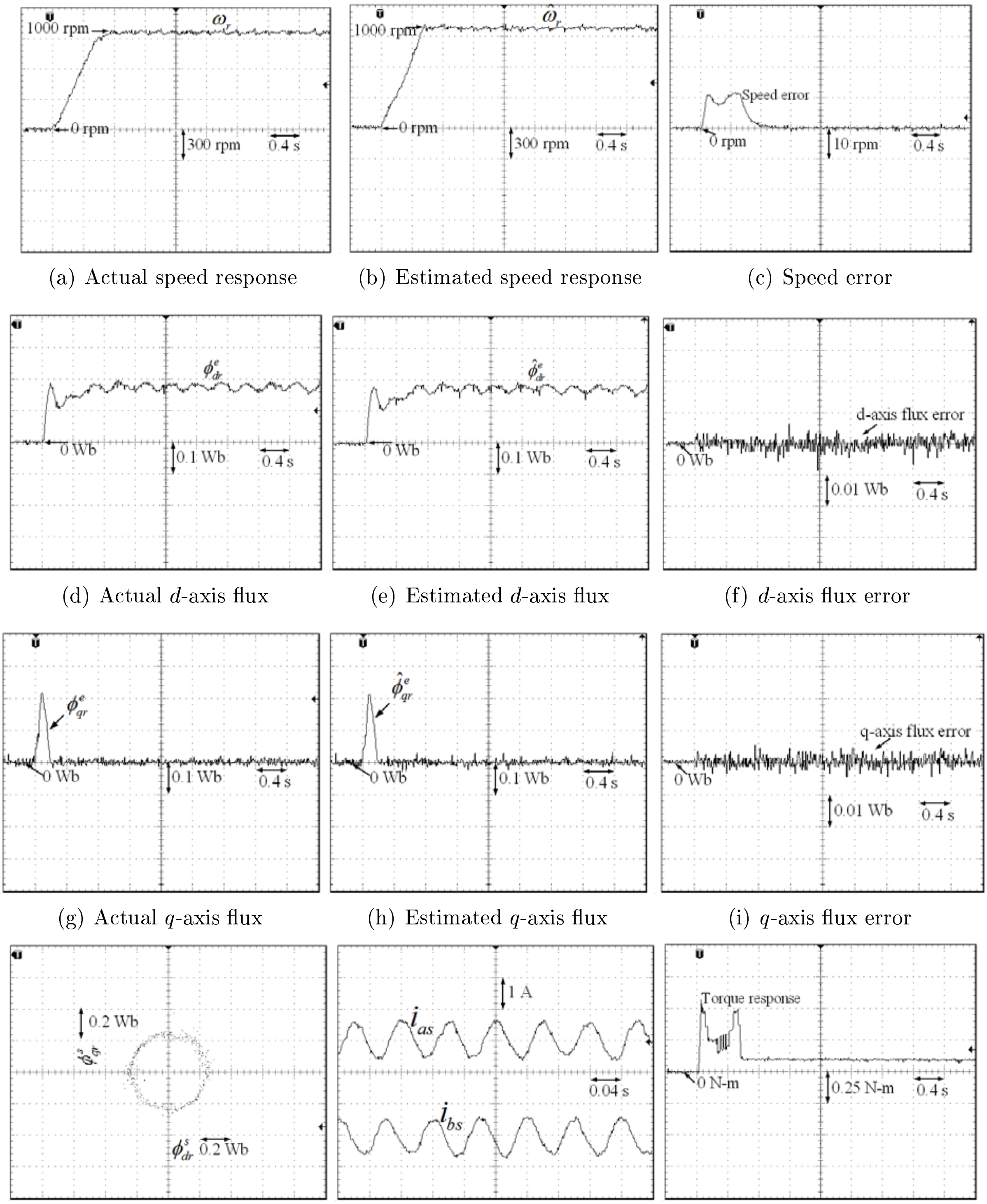
(d) Actual *d*-axis flux (e) Estimated *d*-axis flux (f) The *d*-axis flux error

(g) Actual *q*-axis flux (h) Estimated *q*-axis flux (i) The *q*-axis flux error

(j) Rotor flux locus in *d*-*q* frame (k) Phase-a and phase-b stator currents (l) Torque response

Figure 5. Speed tracking control results for a speed command of 200 rpm with no load

good high speed command response performance. The experimental results show that the proposed algorithm has fairly good performance at both lower and higher speeds without loading.



(a) Actual speed response (b) Estimated speed response (c) Speed error

(d) Actual d-axis flux (e) Estimated d-axis flux (f) d-axis flux error

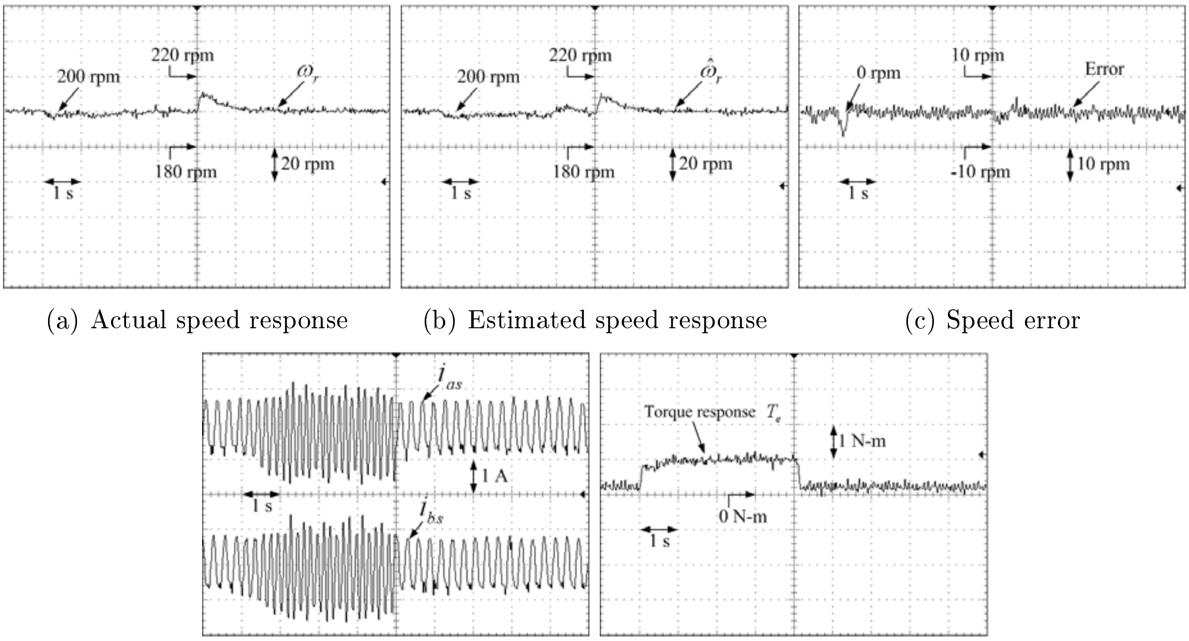
(g) Actual q-axis flux (h) Estimated q-axis flux (i) q-axis flux error

(j) Rotor flux locus in d-q frame (k) Phase-a and phase-b stator currents (l) Torque response

FIGURE 6. Speed tracking control results for a speed command of 1000 rpm with no load

c. Speed regulation control at 200 rpm with loading of 1 N-m

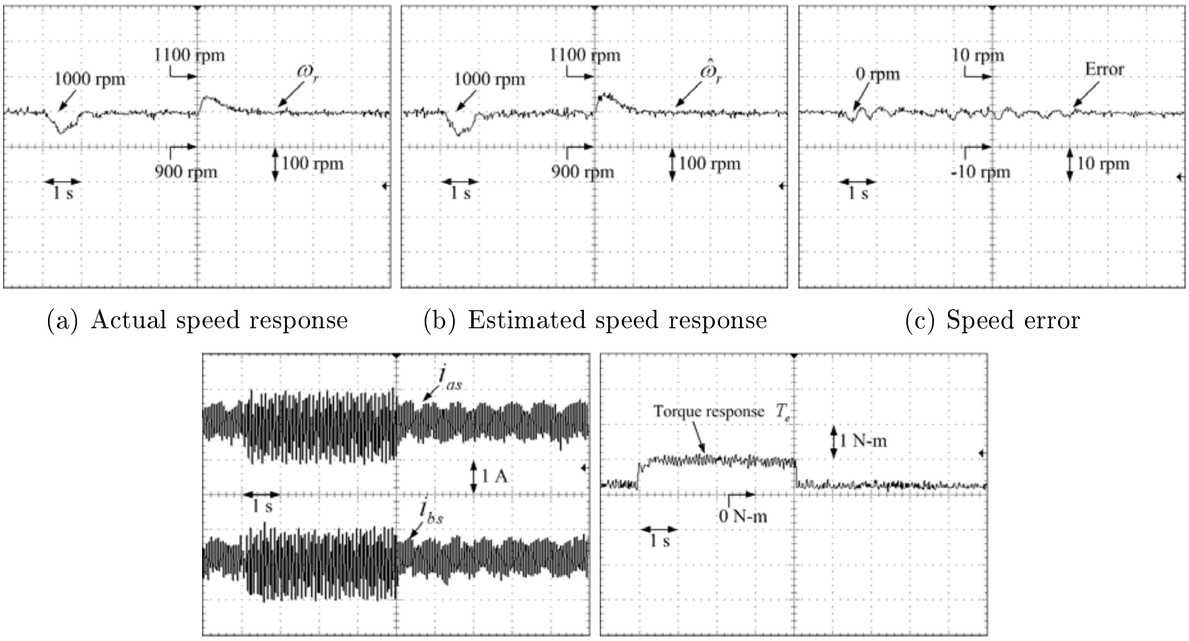
For a speed command of 200 rpm, a load of 1 N-m is added the 1^{st} second and is removed at the 5^{th} second. The actual speed response, estimated speed response and speed error between the actual and estimated speeds are shown in Figures 7(a)-7(c). One



(a) Actual speed response (b) Estimated speed response (c) Speed error

(d) Phase-a and phase-b stator currents (e) Torque response

Figure 7. Speed regulation control results for a speed command of 200 rpm with a load of 1 *N-m*



(a) Actual speed response (b) Estimated speed response (c) Speed error

(d) Phase-a and phase-b stator currents (e) Torque response

Figure 8. Speed regulation control results for a speed command of 1000 rpm with a load of 1 *N-m*

can see in Figures 7(a) to 7(c), that both the actual and estimated speeds can recover to the speed command quickly when the load is added and the estimated speed can also identify the actual speed. At the instant the load is added and removed, the speed response has variations of 5∼10 rpm.

Figure 7(d) presents the phase-a and phase-b stator currents. Figure 7(e) shows the torque response. One can see in Figures 7(d) and 7(e) that during the load period the current and torque magnitudes become larger than the free load time.

d. Speed regulation control at 1000 rpm with loading of 1 *N-m*

For a speed command of 200 rpm, a load of 1 N-m is added the 1^{st} second and is removed at the 5^{th} second. The actual speed response, estimated speed response and speed error between the actual speed and estimated speed are shown in Figures 8(a)-8(c). One can see in Figures 8(a)-8(c), both the actual and estimated speeds can recover to the speed command quickly when the loading is added and the estimated speed can also identify the actual speed.

Figure 8(d) presents the phase-a and phase-b stator currents. Figure 8(e) shows the torque response. One can see in Figures 8(d) and 8(e) during the load period the current and torque magnitudes become larger than the free load time. The experimental results show that the proposed algorithm has fairly good performance in lower speed and higher speed with loading.

5. Conclusions.

This research developed fuzzy neural network speed estimation for induction motor speed control. The steepest descent algorithm was used to adjust the parameters of a four-layer fuzzy neural network that enables precise rotor speed estimation. The proposed fuzzy neural network speed estimation method was implemented in TMS320F2808 DSP. The experimental results proved that the proposed fuzzy neural network speed estimation method is practical and the performance is great.

REFERENCES

[1] R. J. Storey, T. A. Coombs, A. M. K. Campbell, R. A. Weller and D. A. Cardwell, Development of superconducting DC machines using bulk YBCO, *IEEE Trans. Applied Superconductivity*, vol.9, no.2, pp.1253-1256, 1999.

[2] A. Pagel, A. S. Meyer and C. F. Landy, The design of equalizer windings for lap-wound DC machines, *IEEE Trans. Industry Applications*, vol.37, no.2, pp.1000-1011, 2000.

[3] S. J. Chapman, *Electric Machinery Fundamentals*, 3rd Edition, McGraw-Hill, New York, 1999.

[4] H. J. Kim, H. D. Lee and S. K. Sul, A new PWM strategy for common-mode voltage reduction in neutral-point-clamped inverter-fed AC motor drives, *IEEE Trans. Industry Applications*, vol.37, no.6, pp.1840-1845, 2001.

[5] C. D. Angelo, G. Bossio, J. Solsona, G. O. Garcia and M. I. Valla, Mechanical sensorless speed control of permanent-magnet AC motors driving an unknown load, *IEEE Trans. Industrial Electronics*, vol.53, no.2, pp.406-414, 2006.

[6] A. M. Garcıa, T. A. Lipo and D. W. Novotny, A new induction motor V/f control method capable of high-performance regulation at low speeds, *IEEE Trans. Industry Applications*, vol.34, no.4, pp.813-821, 1998.

[7] P. D. C. Perera, F. Blaabjerg, J. K. Pedersen and P. Thogersen, A sensorless stable V/f control method for permanent-magnet synchronous motor drives *IEEE Trans. Industry Applications*, vol.39, no.3, pp.783-791, 2003.

[8] G. K. Singh, D. K. P. Singh, K. Nam and S. K. Lim, A simple indirect field-oriented control scheme for multiconverter-fed induction motor, *IEEE Trans. Industrial Electronics*, vol.52, no.6, pp.1653-1659, 2005.

[9] A. Consoli, G. Scarcella and A. Testa, Slip-frequency detection for indirect field-oriented control drives, *IEEE Trans. Industry Applications*, vol.40, no.1, pp.194-201, 2004.

[10] O. Barambones, P. Alkorta and J. M. G. de Durana, Sliding mode position control for real-time control of induction motors, *International Journal of Innovative Computing, Information and Control*, vol.9, no.7, pp.2741-2754, 2013.

[11] G. Diana and R. G. Harley, An aid for teaching field oriented control applied to induction machines, *IEEE Trans. Power System*, vol.4, no.3, pp.1258-1286, 1989.

[12] Y. A. Kwon and S. H. Kim, A new scheme for speed-sensorless control of induction motor, *IEEE Trans. Industrial Electronics*, vol.51, no.3, 2004.

[13] J. Salomaki, M. Hinkkanen and J. Luomi, Sensorless control of induction motor drives equipped with inverter output filter, *IEEE Trans. Industrial Electronics*, vol.53, no.4, 2006.

[14] T. S. Kwon, M. H. Shin and D. S. Hyun, Speed sensorless stator flux-oriented control of induction motor in the field weakening region using Luenberger observer, *IEEE Trans. Power Electronics*, vol.20, no.4, pp.864-869, 2005.

[15] Y. A. Kwon and S. H. Kim, A new scheme for speed-sensorless control of induction motor, *IEEE Trans. Industrial Electronics*, vol.51, no.3, pp.545-550, 2004.

[16] M. Saejia and S. Sanwongwanich, Averaging analysis approach for stability analysis of speed-sensorless induction motor drives with stator resistance estimation, *IEEE Trans. Industrial Electronics*, vol.53, no.1, pp.162-177, 2006.

[17] Y. R. Kim, S. K. Sul and M. H. Park, Speed sensorless vector control of induction motor using extended Kalman filter, *IEEE Trans. Industry Applications*, vol.30, no.5, pp.1125-1233, 1994.

[18] L. Zhen and L. Xu, Sensorless field orientation control of induction machines based on a mutual MRAS scheme, *IEEE Trans. Industrial Electronics*, vol.45, no.5, pp.824-831, 1998.

[19] C. Lascu, I. Boldea and F. Blaabjerg, Direct torque control of sensorless induction motor drives: A sliding-mode approach, *IEEE Trans. Industry Applications*, vol.40, no.2, pp.582-590, 2004.

[20] T. C. Chen and C. H. Yu, Motion control with deadzone estimation and compensation using GRNN for TWUSM drive system, *Expert Systems with Applications*, vol.36, pp.10931-10941, 2009.

[21] T. C. Chen, T. J. Ren and Y. W. Lou, Ultrasonic motor control based on recurrent fuzzy neural network controller and general regression neural network controller, *Computational Intelligence*, vol.465, pp.291-305, 2013.

[22] J. Peng and R. Dubay, Identification and adaptive neural network control of a DC motor system with dead-zone characteristics, *ISA Transactions*, vol.50, pp.588-598, 2011.

[23] Z. G. Wu, P. Shi, H. Su and J. Chu, Sampled-data fuzzy control of chaotic systems based on a T-S fuzzy model, *IEEE Trans. Fuzzy Systems*, vol.22, no.1, pp.153-163, 2014.

[24] S. E. Shafiei and M. R. Soltanpour, Neural network sliding-mode-PID controller design for electrically driven robot manipulators, *International Journal of Innovative Computing, Information and Control*, vol.7, no.2, pp.511-524, 2011.

[25] Q. Shen, B. Jiang and P. Shi, Adaptive fault diagnosis for T-S fuzzy systems with sensor faults and system performance analysis, *IEEE Trans. Fuzzy Systems*, vol.22, no.2, pp.274-285, 2014.

[26] Z. Wu, P. Shi, H. Su and J. Chu, Stochastic synchronization of Markovian jump neural networks with time-varying delay using sampled data, *IEEE Trans. Cybernetics*, vol.43, no.6, pp.1796-1806, 2013.

[27] S. H. Kim, T. S. Park, J. Y. Yoo and G. T. Park, Speed-sensorless vector control of an induction motor using neural network speed estimation, *IEEE Trans. Industrial Electronics*, vol.48, no.3, pp.609-614, 2001.

[28] T. C. Chen, T. J. Ren and Y. W. Lou, Design of recurrent fuzzy neural network and general regression neural network controller for traveling-wave ultrasonic motor, *International Conference on Neural Computation Theory and Applications*, Paris, France, 2011.

[29] B. K. Bose, *Power Electronics and AC Drives*, Prentice-Hall, Englewood Cliffs, 1986.