Open access • Journal Article • DOI:10.1109/TIA.2003.821797

Development and implementation of a hybrid intelligent controller for interior permanent-magnet synchronous motor drives — Source link

Mohammad Nasir Uddin, M. A. Abido, M.A. Rahman

Institutions: Lakehead University, King Fahd University of Petroleum and Minerals, Memorial University of Newfoundland

Published on: 10 Dec 2002 - IEEE Industry Applications Society Annual Meeting

Topics: Open-loop controller, Control theory, PID controller, Vector control and Synchronous motor

Development and Implementation of a Hybrid Intelligent Controller for Interior Permanent-Magnet Synchronous Motor Drives

M. Nasir Uddin, *Member, IEEE*, M. A. Abido, *Member, IEEE*, and M. Azizur Rahman, *Fellow, IEEE*

***Abstract*—A hybrid neuro-fuzzy scheme for online tuning of a genetic-based proportional–integral (PI) controller for an interior permanent-magnet synchronous motor (IPMSM) drive is presented in this paper. The proposed controller is developed for accurate speed control of the IPMSM drive under various system disturbances. In this work, initially different operating conditions are obtained based on motor dynamics incorporating uncertainties. At each operating condition a genetic algorithm is used to optimize the PI controller parameters in a closed-loop vector control scheme. In the optimization procedure a performance index is developed to reflect the minimum speed deviation, minimum settling time and zero steady-state error. A fuzzy basis function network (FBFN) is utilized for online tuning of the PI controller parameters to ensure optimum drive performance under different disturbances. The proposed FBFN-based PI controller provides a natural framework for combining numerical and linguistic information in a uniform fashion. The proposed controller is successfully implemented in real time using a digital signal processor board DS 1102 for a laboratory 1-hp IPMSM. The effectiveness of the proposed controller is verified by simulation as well as experimental results at different dynamic operating conditions. The proposed controller is found to be robust for applications in an IPMSM drive.**

***Index Terms*—Digital signal processor (DSP), genetic algorithm (GA), neuro-fuzzy control, permanent-magnet synchronous motor, proportional–integral (PI) controller, speed control, vector control.**

I. Introduction

RECENT developments in power semiconductor technology, digital electronics, magnetic materials, and control algorithms have enabled modern ac motor drives to face challenging high-efficiency and high-performance requirements in the industrial sector. Among ac drives, the permanent-magnet synchronous motor has been becoming popular owing to its high torque to current ratio, large power to weight ratio, high efficiency, high power factor, and robustness [1]. These features are due to the incorporation of high energy rare-earth alloys such as neodymium–iron–boron in its construction. In particular, the interior permanent-magnet synchronous motor (IPMSM), which has magnets buried in the rotor core, exhibits certain good properties, such as robust rotor construction, nonsalient rotor, and small effective air gap. The rotors of these machines have complex geometry to ensure optimal use of costly permanent-magnet materials while maintaining high magnetic field in the air gap.

Fast and accurate speed response, quick recovery of speed from any disturbances, and insensitivity to parameter variations are some of the important criteria of the high-performance drive (HPD) systems used in robotics, rolling mills, machine tools, etc. In order to achieve high performance, the vector control of the IPMSM drive is employed [2]. However, the controller design of such a system plays crucial role in the system performance. The decoupling characteristics of a vector-controlled IPMSM are adversely affected by the parameter changes. Traditionally, these control issues are handled by the conventional proportional–integral (PI) controller and other controllers such as the model reference adaptive controller, sliding-mode controller, and variable-structure controller have been widely utilized as speed controllers in the IPMSM drive. However, the difficulties of obtaining the exact d–q-axis reactance parameters of the IPMSM leads to a cumbersome design approach for these controllers. Moreover, the conventional fixed-gain PI controller is very sensitive to step change of command speed, parameter variations, and load disturbances [3]. Again, precise speed control of an IPMSM drive becomes a complex issue due to nonlinear coupling among its winding currents and the rotor speed as well as the nonlinearity present in the electromagnetic developed torque due to magnetic saturation of the rotor core [4]. Therefore, there exists a need to tune the PI controller parameters online to ensure optimum drive performance over a wide range of operating conditions.

The use of an artificial neural network (ANN) alone to design a controller for an IPMSM drive might be insufficient, if the test inputs used to generate training input/output pairs are not rich enough to excite all modes of the system. On the other hand, fuzzy logic controllers (FLCs) are subjective and somewhat heuristic. In most cases, the determination of fuzzy rules, input and output scaling factors, and the choice of membership functions depend on trial and error that makes the design of an FLC a time-consuming task [5]. However, it is important to clarify that the proposed approach brings the learning

Paper MSDAD-A 03–11, presented at the 2002 Industry Applications Society Annual Meeting, Pittsburgh, PA, October 13–18, and approved for publication in the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS by the Industrial Automation and Control Committee of the IEEE Industry Applications Society. Manuscript submitted for review October 15, 2002 and released for publication August 26, 2003.

M. N. Uddin is with the Modeling, Automation and Control Engineering Laboratory, Department of Electrical Engineering, Lakehead University, Thunder Bay, ON P7B 5E1, Canada (e-mail: mnuddin@mail.lakeheadu.ca).

M. A. Abido is with the Electrical Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia.

M. A. Rahman is with the Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John’s, NF A1B 3X5, Canada (e-mail: rahman@engr.mun.ca).

Digital Object Identifier 10.1109/TIA.2003.821797

capabilities of the ANN to the robustness of fuzzy logic systems in the sense that the fuzzy logic concepts are embedded in the network structure and its operation. It also provides a natural framework for combining both numerical information in the form of input/output pairs and linguistic information in the form of IF–THEN rules in a uniform fashion and overcomes the drawbacks of the ANN and FLC. Recently, some work has been reported on the neuro fuzzy techniques for control applications [6]–[9]. However, most of this work is for dc motor or induction motor applications. Moreover, the approach used in this work is different from previous work [6]–[9]. In the present work, a fuzzy basis function network (FBFN) is utilized for speed control of an IPMSM to be used in HPD applications.

II. Motor Dynamics

The mathematical model of an IPMSM drive can be described by the following equations in a synchronously rotating rotor d–q reference frame as

$$\begin{bmatrix} V_d \\ V_q \end{bmatrix} = \begin{bmatrix} R + pL_d & -P\omega_r L_q \\ P\omega_r L_d & R + pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ P\omega_r \psi_f \end{bmatrix} \quad (1)$$

$$T_e = T_L + J_m p\omega_r + B_m\omega_r \quad (2)$$

$$T_e = \frac{3P}{2}\left(\psi_f i_q + (L_d - L_q) i_d i_q\right) \quad (3)$$

where v_d and v_q are the d- and q–axes stator voltages, respectively; i_d and i_q are the d- and q–axes stator currents, respectively; R is the stator resistance per phase; L_d and L_q are the d- and q–axes stator inductances, respectively; T_e and T_L are the electromagnetic and load torques, respectively; J_m is the moment of inertia of the motor and load; B_m is the friction coefficient of the motor; P is the number of poles; ω_r is the rotor speed; p is the differential operator (d/dt); and ψ_f is the rotor magnetic flux linking the stator.

III. Control Principle

The objective of this paper is to obtain the pulsewidth-modulation (PWM) output voltages for the IPMSM drive in order to achieve high-performance speed tracking. For the sake of testing the proposed new technique, the speed control over the normal mode of operation can be achieved by controlling the q-axis component i_q of the supply current as long as the d-axis current i_d is maintained at zero. The simplified resultant model for the IPMSM drive can be represented as [19]

$$pi_q = \frac{1}{L_q}(v_q - Ri_q - P\omega_r\psi_f) \quad (4)$$

$$v_d = -P\omega_r L_q i_q \quad (5)$$

$$T_e = T_L + J_m p\omega_r + B_m\omega_r \quad (6)$$

$$T_e = \frac{3P}{2}(\psi_f i_q). \quad (7)$$

For extended overspeed and constant-power modes of operation the flux-weakening techniques have been used earlier [10]–[13], [19]–[21]. For the maximum torque per ampere mode of operation, the relation between the i_d and i_q currents covering extended overspeed ranges is given as [13],

TABLE I
Machine Parameters

Motor rated power	1 hp
Rated voltage	208 V, 3-phase
Rated current	3 A
Rated frequency	60 Hz
Pole pair number (P)	2
d-axis inductance, L_d	42.44 mH
q-axis inductance, L_q	79.57 mH
Stator resistance, R	1.93 Ω
Motor inertia, J_m	0.003 kgm^2
Friction coefficient, B_m	0.001 Nm/rad/sec
Magnetic flux constant, ψ_f	0.311 volts/rad/sec

$$i_d = \frac{\psi_f}{2(L_q - L_d)} - \sqrt{\frac{\psi_f^2}{4(L_q - L_d)^2} + i_q^2}. \quad (8)$$

It is evident from (4)–(8) that controlling the$d$–q-axes components of the stator current can vary the motor speed. Equations (4)–(8) are used in order to test the proposed FBFN controller over the rated speed while maintaining the maximum torque. The flux-weakening technique is not incorporated to avoid further complexity in real-time implementation. First, various operating conditions are generated randomly by taking into account different drive uncertainties based on (1)–(8) and the machine parameters given in Table I. At each operating condition, a genetic algorithm (GA) is used to optimize PI controller parameters in a closed-loop vector control scheme, which was done offline. In the optimization procedure a performance index is developed in order to ensure minimum speed deviation, minimum settling time, and zero steady-state error. Then, to get an optimum drive performance an FBFN is utilized for online tuning of the PI controller parameters whose initial values are already optimized by the GA. For the sake of completeness the pertinent GA and FBFN are briefly described.

IV. GAs

GAs are exploratory search and optimization procedures that were devised on the principles of natural evolution and population genetics [14]. Unlike other optimization techniques, GAs work with a population of individuals represented by bit strings and modify the population with random search and competition. The advantages of the GA over other traditional optimization techniques can be summarized as follows.

1) GAs search the problem space using a population of trials representing possible solutions to the problem, not a single point. This property ensures that the GA will be less susceptible to getting trapped on local minima.
2) GAs use a performance index assessment to guide the search in the problem space.
3) GAs use probabilistic rules to make decisions.

The GA includes operations such as reproduction, crossover, and mutation. Reproduction is a process in which a new generation of population is formed by selecting the fittest individuals in the current population. Crossover is the most dominant operator in a GA. It is responsible for producing new offspring by selecting two strings and exchanging portions of their structures. The new offspring may replace the weaker individuals in the population. Mutation is a local operator, which is applied with a very low probability. Its function is to alter the value of random position in a string.

A. Real-Coded GA (RCGA)

Due to difficulties of binary representation when dealing with a continuous search space with large dimensions, the proposed approach has been implemented using an RCGA. A decision variable x_i is represented by a real number within its lower limit a_i and upper limit b_i, i.e., $x_i \in [a_i, b_i]$.

The RCGA crossover and mutation operators are described as follows.

1) Crossover: A blend crossover operator has been used in this study. This operator starts by choosing randomly a number from the interval $[x_i - \alpha(y_i - x_i), y_i + \alpha(y_i - x_i)]$, where x_i and y_i are the ith parameter values of the parent solutions and $x_i < y_i$. In order to ensure balance between exploitation and exploration of the search space, $\alpha = 0.5$ is selected.

2) Mutation: The nonuniform mutation operator has been employed in this study. In this operator, the new value x_i' of the parameter x_i after mutation at generation t is given as

$$x_i' = \begin{cases} x_i + \Delta(t, b_i - x_i), & \text{if } \tau = 0 \\ x_i - \Delta(t, x_i - a_i), & \text{if } \tau = 1 \end{cases} \tag{9}$$

and

$$\Delta(t, y) = y\left(1 - r^{\left(1 - \frac{t}{g_{\max}}\right)^{\beta}}\right) \tag{10}$$

where τ is a binary random number, random number $r \in [0, 1]$, g_{max} is the maximum number of generations, and β is a positive constant chosen arbitrarily. In this study, $\beta = 5$ was selected. This operator gives a value $x_i' \in [a_i, b_i]$ such that the probability of returning a value close to x_i increases as the algorithm advances. This makes uniform search in the initial stages where t is small and local at latter stages.

For the optimal settings of PI controller parameters, the following quadratic performance index J is considered:

$$J = \sum_{k=1}^{L} \left[kT_s \Delta\omega(k)\right]^2. \tag{11}$$

In (11), the speed deviation $\Delta\omega(k)$ is weighted by the respective time kT_s. The index J is selected because it ensures small settling time, small steady-state error, and small overshoots. The tuning parameters are adjusted so as to minimize the index J.

B. Computational Flow

Application of the GA for optimal design of the PI controller involves repetitively performing the following two steps.

1) The objective function value must be calculated for each of the strings in the current population. To do this, the FLC parameters must be decoded from each string in the population and the system is simulated to obtain the objective function value.
2) GA operations are applied to produce the next generation of the strings.

These two steps are repeated from one generation to another until the population has converged. The computational flow of the optimization problem can be shown in Fig. 1.

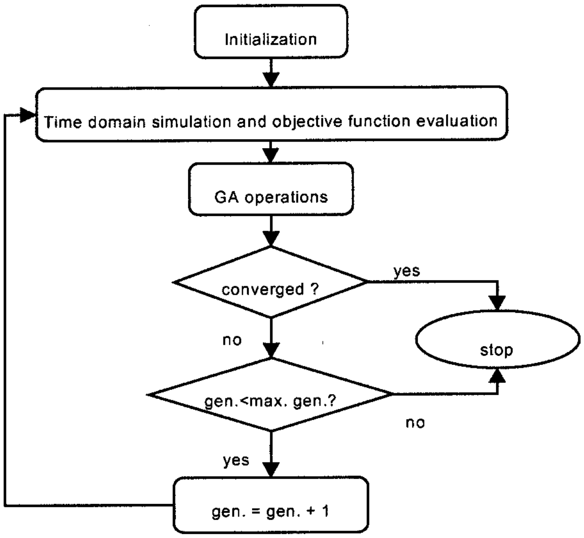


Fig. 1. Computational flowchart of GA.

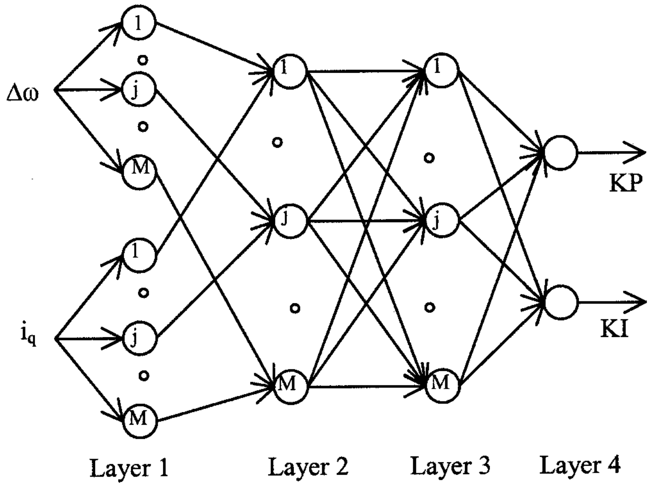


Fig. 2. Schematic diagram of FBFN.

V. FBFN

A. Structure

The specific FBFN has four layers as shown in Fig. 2 [15], [16]. In what follows, we will denote the output of the ith node in the kth layer by $O_i{}^k$. The operation of the network with $n$inputs and m outputs can be described as follows:

Layer 1: For the ith input, every node in this layer computes the degree of membership of the input. Every node j has a function of

$$O_j^1 = \mu_{ij}(x_i), \qquad j = 1, 2, \dots, M \tag{12}$$

where $\mu_{ij}(x_i)$ is a Gaussian membership function associated with ith input and jth rule. It can be expressed as

$$\mu_{ij}(x_i) = \exp\left(-\frac{1}{2}\left(\frac{x_i - c_{ij}}{\sigma_{ij}}\right)^2\right) \tag{13}$$

where c_{ij} and σ_{ij} are the mean and the variance of the jth function.

Layer 2: Every node in this layer multiplies the incoming signals and sends the product out, i.e.,

$$O_j^2 = \prod_{i=1}^{n} \mu_{ij}(x_i), \qquad j = 1, 2, \dots, M. \tag{14}$$

Each node output yields the firing strength of a fuzzy rule.

Layer 3: Every node in this layer calculates the ratio of the jth rule's firing strength to the sum of all rules' firing strengths

$$O_j^3 = \frac{\prod_{i=1}^{n} \mu_{ij}(x_i)}{\sum_{j=1}^{M} \prod_{i=1}^{n} \mu_{ij}(x_i)}, \qquad j = 1, 2, \dots, M. \tag{15}$$

The nodes in this layer compute the normalized firing strength of each rule. The output of each node in this layer represents a fuzzy basis function $p_j(x)$ as

$$p_j(x) = O_j^3, \qquad j = 1, 2, \dots, M \tag{16}$$

where $x = [x_1, \dots, x_n]^T$ is the input vector.

Layer 4: In this layer each node represents an output and linearly combines the fuzzy basis functions as

$$O_k^4 = \sum_{j=1}^{M} p_j(\boldsymbol{x})\theta_{jk}, \qquad k = 1, 2, \dots, m \tag{17}$$

where θ_{jk} is the weight between the jth node in layer 3 and the kth node in layer 4. For learning purposes, the orthogonal least-square (OLS) method is used in this work [16].

B. OLS Learning Algorithm

The objectives of training in this paper are to construct a parsimonious and adequate model of the network, to select a set of appropriate centers $\boldsymbol{c}_j s$ of the hidden units, and to estimate the weights $\theta_{jk} s$. The training input–output pairs are in the form of $\{\boldsymbol{x}(t), \boldsymbol{d}(t)\}$, $t = 1, 2, \dots, N$, where N is the number of training patterns and $\boldsymbol{d}(t) = [d_1(t), \dots, d_m(t)]^T$ is the desired output vector. Initially, all the training data $\{\boldsymbol{x}(t)\}$ are considered as candidates for centers. Initial number of centers M is equal to N. The network output in (12) can be considered as a special case of linear regression model as

$$d_k(t) = \sum_{j=1}^{M} p_j(t)\theta_{jk} + e_k(t), \qquad k = 1, 2, \dots, m \tag{18}$$

where $p_j(t)$ are known as regressors which are fixed functions of the input vector $\boldsymbol{x}(t)$ as

$$p_j(t) = p_j(\boldsymbol{x}(t)) \tag{19}$$

and $e_k(t)$ are the errors between the kth desired and network outputs which are assumed to be uncorrelated with the regressors. By defining

$$\boldsymbol{d_i} = [d_i(1) \dots d_i(N)]^T, \qquad i = 1, 2, \dots, m \tag{20}$$

$$\boldsymbol{e}_i = [e_i(1) \dots e_i(N)]^T, \qquad i = 1, 2, \dots, m \tag{21}$$

$$\boldsymbol{p_j} = [p_j(1) \dots p_j(N)]^T, \qquad j = 1, 2, \dots, M. \tag{22}$$

For $t = 1, 2, \dots, N$ (18) can be expressed as

$$[\boldsymbol{d}_1 \dots \boldsymbol{d}_m][\boldsymbol{p}_1 \cdots \boldsymbol{p}_M] \begin{bmatrix} \theta_{11} & . & . & \theta_{1m} \\ . & & & . \\ . & & & . \\ \theta_{M1} & . & . & \theta_{Mm} \end{bmatrix} + [\boldsymbol{e}_1 \dots \boldsymbol{e}_m] \tag{23}$$

or in matrix form as

$$D = P\Theta + E. \tag{24}$$

The OLS algorithm involves the transformation of the set of $\boldsymbol{p}_j$ into a set of orthogonal basis vectors and uses only the significant ones to form the final FBFN. In general, the number of significant basis vectors in the final network M_s is much less than the initial number M. The regression matrix P can be decomposed as

$$P = WA \tag{25}$$

where A is an $M_s \times M$ upper triangular matrix with unity diagonal elements as

$$A = \begin{bmatrix} 1 & a_{12} & . & a_{1M_s} \\ 0 & 1 & . & . \\ . & . & . & a_{M_s-1M_s} \\ 0 & 0 & . & 1 \end{bmatrix} \tag{26}$$

and W is an $M \times M_s$ matrix with orthogonal columns $\boldsymbol{w}_i$ such that

$$W^T W = H \tag{27}$$

where H is a diagonal matrix.

Using (23), (24) can be rewritten as

$$D = WG + E. \tag{28}$$

The OLS solution for (26) is given by

$$G = H^{-1} W^T D \tag{29}$$

or

$$g_{ij} = \frac{\boldsymbol{w}_i{}^T \boldsymbol{d}_j}{(\boldsymbol{w}_i{}^T \boldsymbol{w}_i)}, \qquad i = 1, 2, \dots, M_s; \quad j = 1, 2, \dots, m. \tag{30}$$

The matrices G and Θ satisfy the triangular system as

$$A\Theta = G. \tag{31}$$

The classical Gram–Schmidt method [16] can be used to derive (31) and, thus, to solve for Θ. The criterion for determining the significance of candidates is the contribution of a candidate to the trace of the desired output covariance matrix. Because the error matrix E is orthogonal to W, it can be shown that the trace of the covariance of $d(t)$ is

$$\text{trace}\left(\frac{D^TD}{N}\right) = \sum_{j=1}^{M_s}\left(\sum_{i=1}^{m} g_{ji}^2\right)\frac{w_j^Tw_j}{N} + \text{trace}\left(\frac{E^TE}{N}\right). \qquad (32)$$

The error reduction ratio due to w_k can be defined as

$$[\text{err}]_k = \left(\sum_{i=1}^{m} g_{ki}^2\right)\frac{w_k^Tw_k}{\text{trace}(D^TD)}, \qquad k = 1,\ldots,M_s. \qquad (33)$$

A candidate regressor is selected at the kth step if it produces the largest value of $[err]_k$ from among the rest candidates. The regressor selection step is given as follows.

```
* At the first step, for $i = 1,2,\ldots,M$ compute:
```

$$\boldsymbol{w}_1^{(i)} = \boldsymbol{p}_i; {g_{1j}}^{(i)} = \frac{\left(\boldsymbol{w}_1^{(i)}\right)^T \boldsymbol{d}_j}{\left(\left(\boldsymbol{w}_1^{(i)}\right)^T \boldsymbol{w}_1^{(i)}\right)}, \qquad j = 1,2,\ldots,m$$

$$[err]_1^{(i)} = \left(\sum_{j=1}^{m}\left({g_{1j}}^{(i)}\right)^2\right)\frac{\left(\boldsymbol{w}_1^{(i)}\right)^T \boldsymbol{w}_1^{(i)}}{trace(D^TD)}$$

```
Find: $[err]_1^{(i1)} = max\{[err]_1^{(i)}, i = 1,2,\ldots,M\}$
Select: $\boldsymbol{w}_1 = \boldsymbol{w}_1^{(i1)} = \boldsymbol{p}_{i1}$; $g_{1j} = {g_{1j}}^{(i1)}$, $j = 1,2,\ldots,m$
* At the kth step where $k = 2,3,\ldots,M_s$, for $i = 1,2,\ldots,M$
* and $i \neq i_1,\ldots,i \neq i_{k-1}$, compute:
```

$${a_{jk}}^{(i)} = \frac{\boldsymbol{w}_j^T \boldsymbol{p}_i}{\left(\boldsymbol{w}_j^T \boldsymbol{w}_j\right)}, \qquad j = 1,2,\ldots,k-1;$$

$$\boldsymbol{w}_k^{(i)} = \boldsymbol{p}_i - \sum_{j=1}^{k-1} {a_{jk}}^{(i)} \boldsymbol{w}_j$$

$${g_{kj}}^{(i)} = \frac{\left(\boldsymbol{w}_k^{(i)}\right)^T \boldsymbol{d}_j}{\left(\left(\boldsymbol{w}_k^{(i)}\right)^T \boldsymbol{w}_k^{(i)}\right)}, \qquad j = 1,2,\ldots,m$$

$${[err]_k}^{(i)} = \left(\sum_{j=1}^{m}\left({g_{kj}}^{(i)}\right)^2\right)\frac{\left(\boldsymbol{w}_k^{(i)}\right)^T \boldsymbol{w}_k^{(i)}}{trace(D^TD)}$$

```
Find:
```

$$[err]_k^{(ik)} = max\left\{[err]_k^{(i)}, i = 1,2,\ldots,M, \; i \neq i_1,\ldots,i \neq i_k - 1,\right\}$$

```
Select $\boldsymbol{w}_k = \boldsymbol{w}_k^{(ik)}$; $g_{kj} = {g_{kj}}^{(ik)}$, $j = 1,2,\ldots,m$
* The procedure is terminated at the $M_s$th step when
```

$$1 - \sum_{j=1}^{M_s} [err]_j < \lambda$$

```
where $0 < \lambda < 1$ is a chosen tolerance.
This gives a subset
model containing $M_s$ significant
regressors.
```

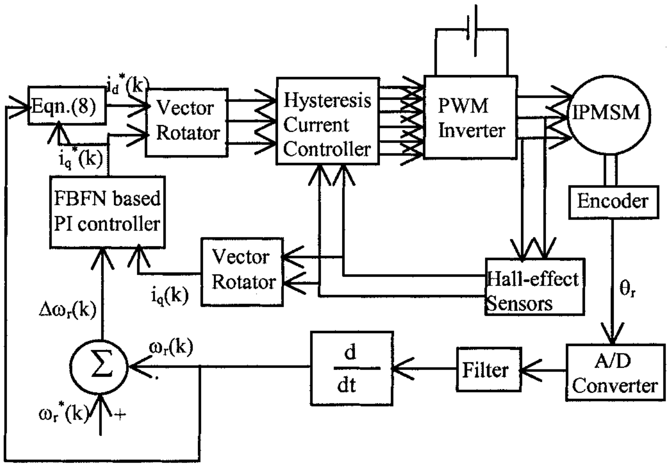


Fig. 3. Block diagram of the proposed FBFN-based controller for IPMSM drive.

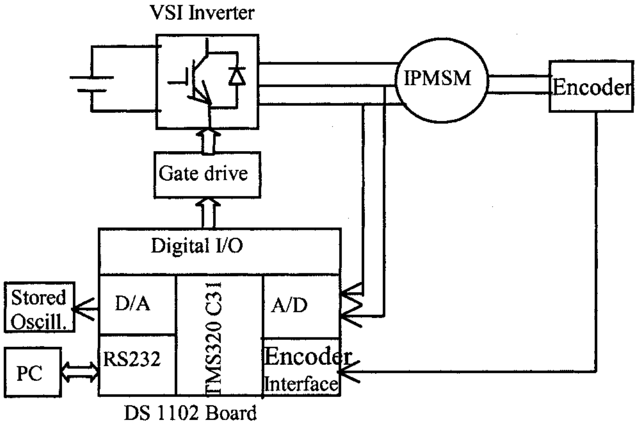


Fig. 4. Hardware schematic for real-time implementation.

VI. Real-Time Implementation

The block diagram of the closed-loop vector control scheme of the IPMSM incorporating the proposed FBFN controller is shown in Fig. 3. The drive is experimentally implemented using a digital signal processor (DSP) board DS1102 through both hardware and software [17].

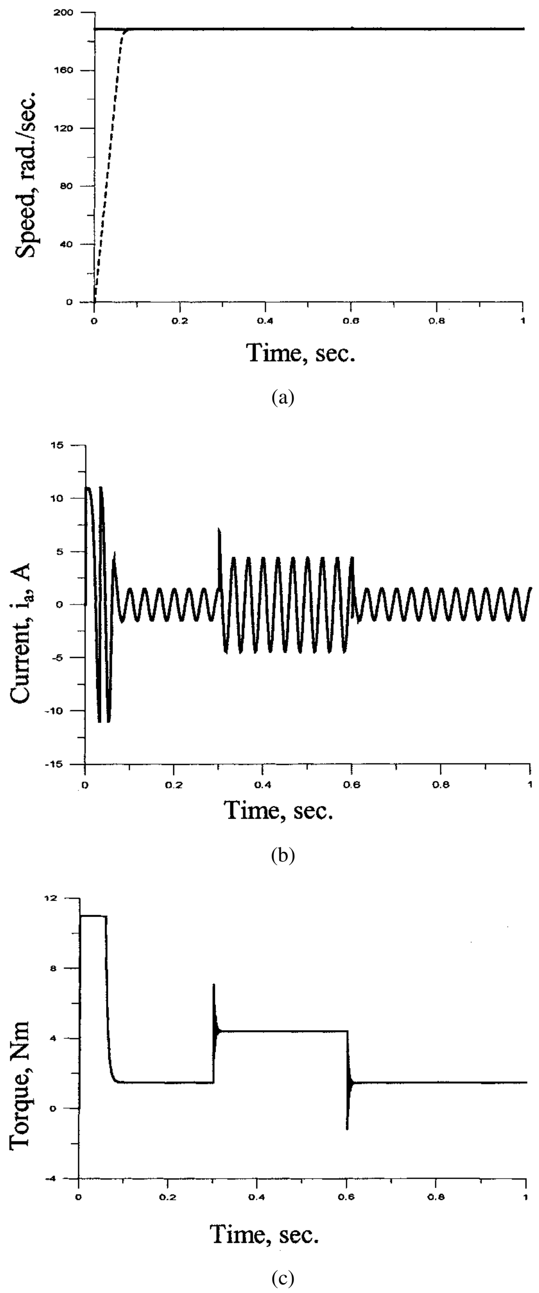
The hardware schematic for real-time implementation of the proposed IPMSM drive is shown in Fig. 4. The DS1102 board is based on a Texas Instruments (TI) TMS320C31, a 32-bit floating-point DSP. The DSP has been supplemented by a set of on-board peripherals used in digital control systems, such as A/D, D/A converters and incremental encoder interfaces. The DS 1102 is also equipped with a TI TMS320P14, a 16-bit microcontroller DSP that acts as a slave processor and is used for some special purposes. In this work, the slave processor is used for digital I/O configuration. The actual motor currents are measured by the Hall-effect sensors, which have good frequency response and are fed to the DSP board through an A/D converter. As the motor neutral is isolated, only two phase currents are fed back and the other phase current is calculated from them. The rotor position is measured by an optical incremental encoder which is mounted at the rotor shaft end. Then, it fed to the DSP board through encoder interface. The encoder generates 4096 pulses per revolution. By using a fourfold pulse multiplication the number of pulses is increased to 4 × 4096 in order to get better resolution. A 24-bit position counter is used to count the encoder pulses and is read by a calling function in the software.

The motor speed is calculated from the rotor position by backward difference interpolation. A digital moving average filter is used to remove the noise from the speed signal. The calculated actual motor speed compared with command speed and the speed error is used to calculate the torque component of the current i_q^* using the FBFN-based PI control algorithm. The command a–b–c phase currents are generated from i_q^* and i_d^* using inverse Park's transformation [5], [17]. In order to implement the vector control algorithm, the hysteresis controller is used as the current controller. It is to be noted that other types of current controllers have also been used earlier [22], [23]. However, it is beyond the scope of this paper. The hysteresis current controller compares the command currents with the corresponding actual motor currents and generates the logic signals, which act as firing pulses for the inverter switches. Thus, these six PWM logic signals are the output of the DSP board and are fed to the base drive circuit of the inverter power module. The D/A channels are used to capture the necessary output signals in a digital storage oscilloscope. The complete IPMSM drive is implemented through software by developing a program in high-level ANSI "C" programming language. The program is compiled by the TI "C" compiler and then the program is downloaded to the DSP controller board. The sampling frequency for experimental implementation of the proposed IPMSM drive system is 6.67 kHz.

VII. Simulation and Experimental Results

Several tests were performed to evaluate the performance of the proposed FBFN-based IPMSM drive system both in simulation and experiment. The speed, stator current, and torque responses are observed under different operating conditions such as sudden change in command speed, step change in load, parameter variations, etc. Sample results are presented below. The complete drive has been simulated using Matlab/Simulink [18].

The simulated motor speed, current, and torque responses are shown in Fig. 5(a)–(c) to see the starting performance as well as the response with a load disturbance of the drive. The drive system is started at a constant load of 1 N·m with the speed



(a)

(b)

(c)

Fig. 5. Simulated starting responses of the proposed hybrid intelligent controller based drive: (a) speed, (b) stator current, i_a, and (c) torque.

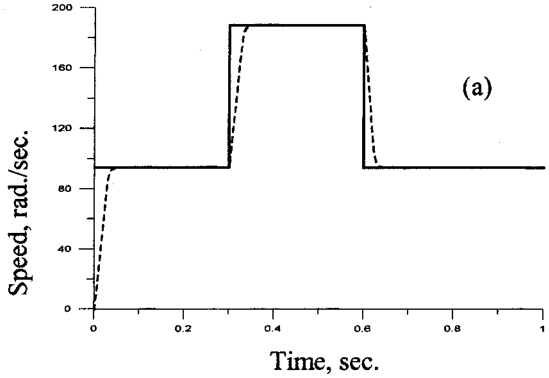


Fig. 6. Simulated speed responses of the drive for a step change in command speed.

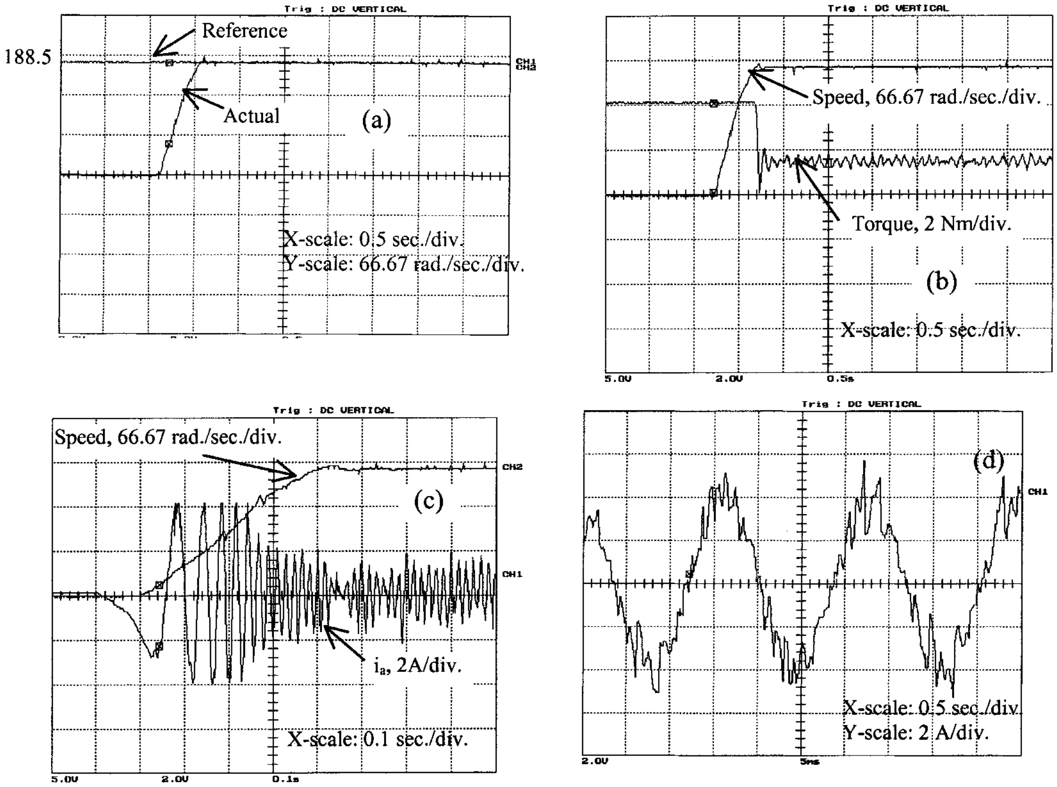


Fig. 7. Experimental starting responses of the proposed drive at full-load condition: (a) speed, (b) torque, (c) stator current i_a, and (d) steady-state current i_a.

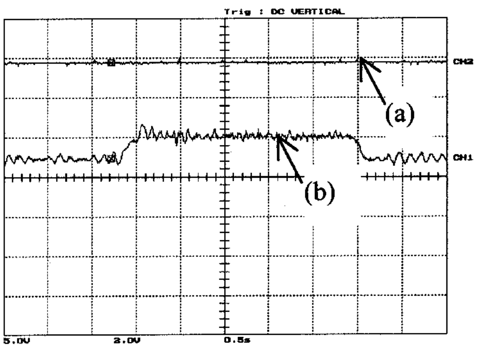


Fig. 8. Experimental responses of the proposed drive for a step change in load: (a) speed (66.67 rad/s/div) and (b) torque (2 N·m/div).

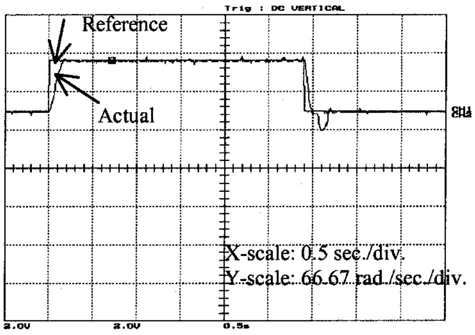


Fig. 9. Experimental speed responses of the proposed drive for a step change in speed.

reference set at 1800 r/min (188.5 rad/s). It can be seen from Fig. 5(a) that the actual speed converges to the reference value within 0.1 s without any overshoot/undershoot and with zero steady-state error. At $t = 0.3$ s, a load torque of 2 N·m is applied to the motor shaft in a stepwise manner. The actual speed does not change during the disturbance while the stator current swiftly reaches its new value corresponding to the load applied. This shows that the new controller is capable of starting from standstill condition to the rated speed as well as rejecting the load disturbance. Another simulated speed response for a sudden change in command speed is shown in Fig. 6. It is evident from Fig. 6 that the proposed FBFN-based drive is also capable of handling the disturbance in speed command.

The experimental starting performance including speed, torque, transient, and steady-state stator current i_a are shown in Fig. (7a)–(d), respectively. It is shown that the proposed drive is also capable of following the command speed very quickly with zero steady-state error and almost without any overshoot/undershoot in a real-time situation. Fig. 8 shows another speed and the corresponding torque responses for a step change in load torque using the dynamometer. It is shown in Fig. 8 that the drive is insensitive with load disturbance. Another experimental speed response is shown in Fig. 9 for a step change in command speed. It is evident that the proposed drive can adapt itself with speed disturbance. Fig. 10 shows the speed response of the proposed drive for a doubled stator resistance. The extra resistances are inserted externally to the stator windings. It is evident from Fig. 10 that the drive can handle the parameter variations smoothly.

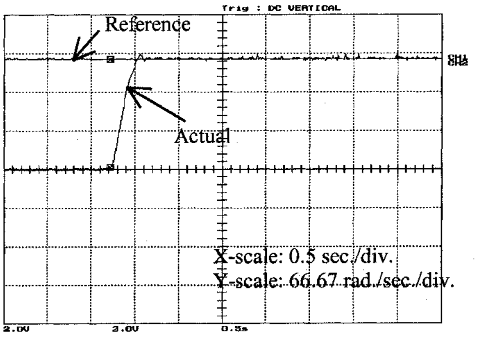


Fig. 10. Experimental speed responses of the proposed drive for a doubled stator resistance.

VIII. Conclusion

A novel speed control technique based on a hybrid intelligent controller for an IPMSM drive has been presented in this paper. In the proposed hybrid intelligent approach a new neuro-fuzzy technique is used for online tuning of the parameters of a PI controller whose initial values are optimized by a GA. The closed-loop vector control of the IPMSM drive incorporating the proposed FBFN controller has been successfully implemented in real time for a laboratory 1-hp IPMSM drive. The PI controller parameters have been optimized offline using a GA with a performance index to reflect the minimum settling time, minimum overshoot/undershoot, and zero steady-state error. Based on the optimized operating conditions and control parameters the FBFN structure has been developed and trained for online tuning of the PI controller parameters. The validity of the proposed control technique has been established both in simulation and experiment at different operating conditions such as sudden load change, step change of speed, parameter variations, etc. There is close agreement between simulation and experimental results. The drive has been found robust in terms of quick response and disturbance rejection.

References

[1] G. R. Slemon, *Electric Machines and Drives*. Reading, MA: Addison-Wesley, 1992, pp. 503–511.

[2] F. Blaschke, “The principle of filed orientation as applied to the new transvector closed-loop control system for rotating-field machines,” *Siemens Rev.*, vol. 34, no. 3, pp. 217–220, May 1972.

[3] M. N. Uddin, T. S. Radwan, and M. A. Rahman, “Performances of fuzzy logic based indirect vector control for induction motor drive,” *IEEE Trans. Ind. Applicat.*, vol. 38, pp. 1219–1225, Sept./Oct. 2002.

[4] M. A. Rahman and P. Zhou, “Field circuit analysis of brushless permanent magnet synchronous motors,” *IEEE Trans. Ind. Electron.*, vol. 43, pp. 256–267, Apr. 1996.

[5] M. N. Uddin and M. A. Rahman, “Fuzzy logic based speed control of an IPM synchronous motor drive,” *J. Adv. Comput. Intell.*, vol. 4, no. 3, pp. 212–219, 2000.

[6] G. S. Buja and F. Todesco, “Neural network implementation of fuzzy logic controller,” *IEEE Trans. Ind. Electron.*, vol. 41, pp. 663–667, Dec. 1994.

[7] E. Cerruto, A. Consoli, A. Raciti, and A. Testa, “Fuzzy adaptive vector control of induction motor drives,” *IEEE Trans. Power Electron.*, vol. 12, pp. 1028–1039, Nov. 1997.

[8] B. K. Bose, N. R. Patel, and K. Rajashekara, “A neuro-fuzzy based on-line efficiency optimization control of a stator flux-oriented direct vector-controlled induction motor drive,” *IEEE Trans. Ind. Electron.*, vol. 44, pp. 270–273, Apr. 1997.

[9] A. Rubaai, D. Ricketts, and M. D. Kankam, “Development and implementation of an adaptive fuzzy-neural-network controller for brushless drives,” *IEEE Trans. Ind. Applicat.*, vol. 38, pp. 441–447, Mar./Apr. 2002.

[10] B. Sneyers, D. W. Novotny, and T. A. Lipo, “Field-weakening in buried permanent magnet AC motor drives,” *IEEE Trans. Ind. Applicat.*, vol. IA-21, pp. 398–407, Mar./Apr. 1985.

[11] T. M. Jahns, “Flux-weakening regime operation of an interior permanent-magnet synchronous motor drive,” *IEEE Trans. Ind. Applicat.*, vol. IA-23, pp. 681–689, July/Aug. 1987.

[12] S. Morimoto, M. Sanada, and Y. Takeda, “Effects and compensation of magnetic saturation in flux-weakening controlled permanent magnet synchronous motor drives,” *IEEE Trans. Ind. Applicat.*, vol. 30, pp. 1632–1637, Nov./Dec. 1994.

[13] M. N. Uddin, T. S. Radwan, and M. A. Rahman, “Performance of interior permanent magnet synchronous motor over wide speed range,” *IEEE Trans. Energy Conversion*, vol. 17, pp. 79–84, Mar. 2002.

[14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[15] L. X. Wang and J. M. Mendel, “Fuzzy basis functions, universal approximation, and orthogonal least-squares learning,” *IEEE Trans. Neural Networks*, vol. 3, pp. 807–814, Sept. 1992.

[16] M. A. Abido and Y. L. Abdel-Magid, “A hybrid neuro-fuzzy power system stabilizer for multi-machine power systems,” *IEEE Trans. Power Syst.*, vol. 13, pp. 1323–1330, Nov. 1998.

[17] *Digital Signal Processing and Control Engineering, Manual Guide*, dSPACE GmbH, Paderborn, Germany, 1996.

[18] *Matlab/Simulink User Guide*, The Math Works Inc., Natick, MA, 1997.

[19] D. W. Novotny and T. A. Lipo, *Vector Control and Dynamics of AC Drives*. New York: Oxford Univ. Press, 1997, ch. 7.

[20] R. Krishnan, *Electric Motor Drives*. Upper Saddle River, NJ: Prentice-Hall, 2001, ch. 9.

[21] B. K. Bose, *Modern Power Electronics and AC Drives*. Upper Saddle River, NJ: Prentice-Hall PTR, 2002, ch. 9.

[22] K. M. Rahman, M. R. Khan, M. A. Choudhury, and M. A. Rahman, “Variable band hysteresis current controllers for PWM voltage-source inverters,” *IEEE Transactions Power Electron.*, vol. 12, pp. 964–970, Nov. 1997.

[23] M. N. Uddin, T. S. Radwan, G. H. George, and M. A. Rahman, “Performance of current controllers for VSI-Fed IPMSM drive,” *IEEE Trans. Ind. Applicat.*, vol. 36, pp. 1531–1538, Nov./Dec. 2000.

M. Nasir Uddin (S’98–M’00) was born in Rajbari, Bangladesh, in 1969. He received the B.Sc. and M.Sc. degrees in electrical and electronic engineering from the Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, in 1993 and 1996, respectively, and the Ph.D. degree in electrical engineering from the Memorial University of Newfoundland (MUN), St. John’s, NF, Canada, in 2000.

He is currently an Assistant Professor with the Department of Electrical Engineering, Lakehead University, Thunder Bay, ON, Canada, where he is engaged in teaching and research. From January 2001 to May 2001, he was an Assistant Professor with the Department of Electrical and Computer Engineering, University of South Alabama, Mobile. From May 2001 to August 2001, he was a Post-Doctoral Fellow with MUN. From 1996 to 1997, he was an Assistant Professor and from 1994 to 1996, he was a Lecturer with the BUET. From 1996 to 2000, he was an Instructor with the College of the North Atlantic, St. John’s, NF, Canada. From September 1997 to August 2000, he was a Teaching Assistant with MUN. His research interests include power electronics, electric motor drives and the application of neural network and fuzzy logic to power apparatus. He has authored or coauthored over 30 papers.

Dr. Uddin is the recipient of the First Prize Paper Award of the Industrial Automation and Control Committee of the IEEE Industry Applications Society.

M. A. Abido (M'98) received the B.Sc. (Honors with first class) and M.Sc. degrees in electrical engineering from Menoufia University, El-Shebin, Egypt, in 1985 and 1989, respectively, and the Ph.D. degree from King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 1997.

He has been with the Electrical Engineering Department, Menoufia University, as a Graduate Assistant (1985–1989), as a Lecturer (1989–1992), and as an Assistant Professor (since 1998). He has also been an Assistant Professor at KFUPM since 1997. His research interests are power system planning, operation, and optimization techniques applied to power systems.

Dr. Abido is the recipient of the First Prize Paper Award of the Industrial Automation and Control Committee of the IEEE Industry Applications Society.

M. Azizur Rahman (S'67–M'68–SM'73–F'88) was born in Santahar, Bangladesh, in 1941. He received the B.Sc. degree in electrical engineering from the Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, in 1962, the M.A.Sc. degree from the University of Toronto, Toronto, ON, Canada, in 1965, and the Ph.D. degree from Carleton University, Ottawa, ON, Canada, in 1968.

In 1962, he joined the Department of Electrical Engineering, BUET, as a Lecturer, and then became an Assistant Professor in 1969, Associate Professor in 1972, and Professor in 1975. In 1976, he joined the Memorial University of Newfoundland (MUN), St. John's, NF, Canada, where is currently a Professor and University Research Professor. He possesses 42 years of teaching including approximately ten years of full-time and concurrent industrial, utility, and consulting experiences at GE, Schenectady, GE Canada, Peterborough, Newfoundland Hydro, Dhaka Electric Supply, Iron Ore Company of Canada, etc. He was a Visiting Research Fellow at the Technische Hogeschool Eindhoven, The Netherlands, in 1973 and 1975, a Nuffield Fellow at Imperial College, London, U.K., from 1974 to 1975, and a Visiting Fellow at the University of Toronto in 1975 and 1984–1985. He has held Visiting Professorships at the Nanyang Technological University (1991–1992, 1999–2000), Tokyo Institute of Technology (1992) and Science University of Tokyo (1999). He has authored or coauthored over 475 papers and five books. He holds 11 patents. His current research interests are machines, power systems, digital protection, and power electronics.

Dr. Rahman is a Registered Professional Engineer in the Provinces of Newfoundland and Ontario, Canada. He is a member of the Institution of Electrical Engineers, Japan, a Fellow of the Institution of Electrical Engineers (IEE), U.K., a Fellow of the Engineering Institute of Canada, and a Life Fellow of the Institution of Engineers, Bangladesh. He has been the recipient of numerous awards, including the GE Centennial Award for Invention Disclosures in 1978, IEEE Outstanding Students' Counselor Award in 1980, IEEE Notable Service Award for contributions in IEEE and Engineering Professions in 1987, IEEE Industry Application Society's Outstanding Achievement Award in 1992, the Association of Professional Engineers Merit Award in 1994, the IEEE Canada Outstanding Engineering Educator's Medal in 1996, IEEE Third Millennium Medal in 2000, and IEEE Power Engineering Society Cyrill Veinott Electromechanical Energy Conversion Award in 2003. He is also the recipient of three First Prize Paper Awards from IEEE Industry Applications Society Committees.