JPE 4-3-1

Hybrid Induction Motor Control Using a Genetically Optimized Pseudo-on-line Method

Jong-seok Lee**, Kyung-won Jang*, J. F. Peters*** and Tae-chon Ahn*†

*School of Electrical & Electronic Eng., Wonkwang University, Chonbuk, Korea
**Department of Radiotechnology, Wonkwang Health Science College, Chonbuk, Korea
***Department of Electrical and Computer Eng., University of Manitoba, Winnipeg, Canada

ABSTRACT

This paper introduces a hybrid induction motor control using a genetically optimized pseudo-on-line method. Optimization results from the use of a look-up table based on genetic algorithms to find the global optimum of an unconstrained optimization problem. The approach to induction motor control includes a pseudo-on-line procedure that optimally estimates parameters of a fuzzy PID (FPID) controller. The proposed hybrid genetic fuzzy PID (GFPID) controller is applied to speed control of a 3-phase induction motor and its computer simulation is carried out. Simulation results show that the proposed controller performs better than conventional FPID and PID controllers. The contribution of this paper is the introduction of a high performance hybrid form of induction motor control that makes on-line and real-time control of the drive system possible.

Keywords: Hybrid; Control; Fuzzy; Genetic; Induction Motor; Optimization; genetically optimized, Pseudo-on-line

1. Introduction

Electric-power vehicles use a variety of motor drive systems. The recent introduction of multi-purpose electric vehicles indicates that both ac induction and brushless DC motors are gaining in popularity for traction motor applications [1]. The induction motor control problem has been widely studied with the objectives of obtaining better results in terms of stability, robustness to parameters variation and disturbances rejection. The voltage or current and frequency are the basic control variables of the induction motor. Many algorithms have been employed to improve the performance of the induction motor control [2], [15].

The 3-phase induction motor is a representative plant, and the conventional PID controllers are used extensively in its control [3],[16]. They are inexpensive and very effective for simple linear systems. Use of these conventional controllers is often adequate when the non-linearity of process is mild and plant operations are constrained to small regions at a nominal steady - state. Model-based non-linear control techniques can be used when high performance is required over a broader range of operating conditions. However, the approaches require an accurate model of the process. A simpler alternative, although with some loss in performance, is to use linear controllers with gain scheduling. The design of

Manuscript received August 5, 2003; revised Mar. 8, 2004.
†Corresponding Author: tcahn@wonkwang.ac.kr, Wonkwang Univ. Tel: +82- 63-850-6344, Fax: +82-63-853-2196
* School of Electrical & Electronic Eng.,Wonkwang Univ.
** Dept. of Radiotechnology Wonkwang Health Science College
*** Dept. of Electrical and Computer Eng Univ. of Manitoba

discrete-time fuzzy PID (FPID) controllers in various combinations results in a new fuzzy version of the result of the conventional PID controllers [4]. These controllers have the same linear structure as the conventional PID controllers in the proportional, integral and derivative parts, but have non-constant gains, namely, the proportional, integral, and derivative gains are nonlinear functions of the input signals. The FPID controllers thus preserve the simple linear structure of the conventional controllers, and yet enhance the self-tuning control capability for non-linearity [5],[6].

In this paper, a novel control method is proposed with the pseudo-on-line scheme that auto-tunes the parameters of a controller by the genetic algorithm [7] that does not use the gradient and finds the global optimum of an un-constrained optimization problem, for the improvement and optimization of systems. This technique includes a pseudo-on-line procedure that optimally estimates off-line parameters of the FPID controller using a genetic algorithm, and makes the optimized look-up table using the estimated parameters and controls in on-line systems with non-linearity at real-time. This method is applied to the FPID controller in the drive system of an induction motor. The proposed hybrid genetic fuzzy PID (GFPID) controller with the auto-tuning function executes the speed control of the system. Authors divide the region of errors that have influence on the system parameters into several error levels and then assign each level based on the optimized look-up table using the genetic algorithm. This makes on-line and real-time control of the drive system of induction motors possible. Computer simulations show that the proposed controller of a fuzzy control algorithm in the transient state and the GA-PID control algorithm in the steady state combine to improve control functions. This enhanced effectively depresses fluctuation of output more than conventional FPID and PID controllers.

2. Induction motor modeling

In this article, induction motor modeling is presented relative to a motor similar to the one shown in Fig. 1.

Since the stator or rotor is assumed to have a symmetrical air gap, it is possible to express its voltage equations of the three-phase induction motor in the stationary coordinates as (1)-(2) [1],[8].

$$v_{abcs} = R_s i_{abcs} + \frac{d\lambda_{abcs}}{dt} \tag{1}$$

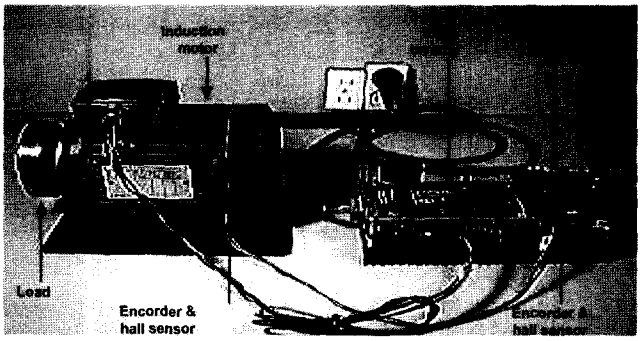


Fig. 1. Sample induction motor configura

$$v_{abcr} = R_r i_{abcr} + \frac{d\lambda_{abcr}}{dt} \tag{2}$$

The *d-q* reference frames are usually selected on the basis of convenience or compatibility with the representations of other network components. Those of the induction machine in the stationary reference frame can be obtained by the reference-frame theory [8].

The *d-q* transformation matrix s_{dq0} is given by:

$$s_{dq0} = \frac{2}{3}\begin{bmatrix} \cos\omega t & \cos(\omega t - \frac{2\pi}{3}) & \cos(\omega t + \frac{2\pi}{3}) \\ \sin(\omega t) & \sin(\omega t - \frac{2\pi}{3}) & \sin(\omega t + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{3}$$

Using (3), the transformation equation from $a-b-c$ to $d-q$ reference frame is as follows:

$$f_{dq0} = s_{dq0} f_{abc} \tag{4}$$

The stator $d-q$ voltage equation in a synchronously rotating frame can be written in terms of component *d* and *q* voltages as follows:

$$v_{qs} = R_s i_{qs} + \frac{d\lambda_{qs}}{dt} + \omega_e \lambda_{ds} \tag{5}$$

$$v_{ds} = R_s i_{ds} + \frac{d\lambda_{ds}}{dt} - \omega_e \lambda_{qs} \tag{6}$$

The rotor $d-q$ voltage equation in a synchronously rotating frame can be also written in terms of component *d* and *q* voltages as follows:

$$v_{qr} = R_r i_{qr} + \frac{d\lambda_{qr}}{dt} + (\omega_e - \omega_r)\lambda_{dr} \qquad (7)$$

$$v_{dr} = R_r i_{dr} + \frac{d\lambda_{dr}}{dt} - (\omega_e - \omega_r)\lambda_{qr} \qquad (8)$$

The stator and rotor flux linkage expressions in terms of the currents can be written compactly as follows [8]:

$$\begin{aligned} \lambda_{qs} &= L_{ls} i_{qs} + L_m (i_{qs} + i_{qr}) \\ \lambda_{ds} &= L_{ls} i_{ds} + L_m (i_{ds} + i_{dr}) \\ \lambda_{qr} &= L_{lr} i_{qr} + L_m (i_{qs} + i_{qr}) \\ \lambda_{dr} &= L_{lr} i_{dr} + L_m (i_{ds} + i_{dr}) \end{aligned} \qquad (9)$$

The induction motor state space modeling, which describes its electric behavior in a $d-q$ synchronous rotating reference frame, is written below in a standard notation.

$$\begin{bmatrix} v_{qs} \\ v_{ds} \\ v_{qr} \\ v_{dr} \end{bmatrix} = \begin{bmatrix} R_s + L_s \frac{d}{dt} & \omega_e L_s & L_m \frac{d}{dt} & \omega_e L_m \\ -\omega_e L_s & R_s + L_s \frac{d}{dt} & -\omega_e L_m & L_m \frac{d}{dt} \\ L_m \frac{d}{dt} & \omega_{sl} L_m & R_r + L_r \frac{d}{dt} & \omega_{sl} L_r \\ -\omega_{sl} L_m & L_m \frac{d}{dt} & -\omega_{sl} L_r & R_r + L_r \frac{d}{dt} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \qquad (10)$$

The expression for the electromagnetic torque in terms of current is as follows:

$$T_e = (\frac{3}{2})(\frac{P}{2}) L_m (i_{qs} i_{dr} - i_{ds} i_{qr}) \qquad (11)$$

Where *P* represents the number of poles, T_e represents positive motor action.

The vector control of the induction motor [3] is the current accepted method for high performance system response requirements. It is based on the decoupling of the magnetizing and torque producing components of the stator current. Under this condition, the *q*-axis component of the rotor current i_{qr} can be set to zero, while the *d*-axis reaches the nominal value of the magnetizing flux. From there, the torque equation is expressed basically as follows:

$$T_e = \frac{3}{2}\frac{P}{2} L_m (i_{qs} i_{dr}) \qquad (12)$$

Solving for i_{qs}, the torque component current i_{qs} can be written as follows:

$$i_{qs} = \frac{2}{3}\frac{2}{P}\frac{1}{L_m}\frac{1}{i_{dr}} T_e \qquad (13)$$

In this paper, the 3-phase induction motor is controlled using the pseudo-on-line method based on genetic algorithms. The pseudo-on-line method based on genetic algorithms contains the following steps:

Step 1. Code errors, integral and derivatives of error.

Step 2. Compute fitness value of the coding parameters.

Step 3. Reproduce and select strings to create new mating pool.

Step 4. Generate new population by crossover and mutation of selection strings.

Step 5. Optimize fuzzy membership functions.

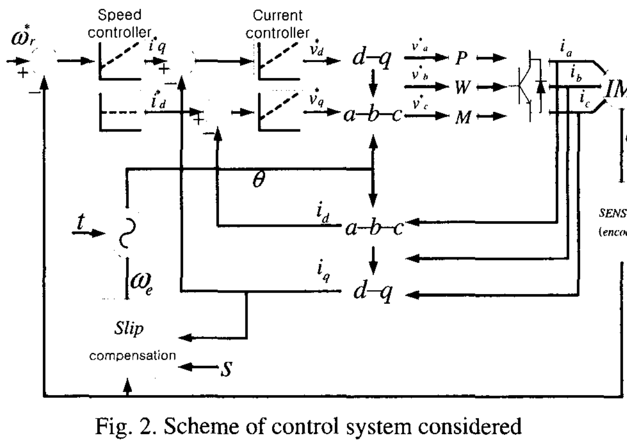


Fig. 2. Scheme of control system considered

The scheme of the control system considered is shown in Fig. 2. In Fig. 2, ω_r is the rotor angular frequency, ω_e is the stator angular frequency, *S* is the slip and θ is the angle. In these designations, i and v denote currents and voltages at *d-q* and $a-b-c$ axes. Mark * refers to the reference value.

3. Direct FPID control algorithm.

In the design and stability analysis of the FPID controller, the approach to determining the control gains that gave acceptable outputs was performed by manually and automatically tuning these gains until a set of gains that yielded satisfactory outputs were obtained [9].

$$u(t) = k_p e(t) + k_d \frac{de(t)}{dt} \qquad \text{<PD>} \qquad (14)$$

$$u = \sum_{i=1}^{k} (\mu(e_i) \cap \mu(de_i)) \qquad \text{<Fuzzy PD>} \qquad (15)$$

Where $\cap$ denotes the intersection of membership function, k_p denotes proportional coefficient, k_d derivative coefficient, $e(t)$ error term, $\mu(e_i)$ membership function of error term, and $\mu(de_i)$ the membership function of the increments of error term.

The PID control input in (14) does not consider the integral of error term. The controller does not show high performance in the steady state. To remedy this problem, a fuzzy control system transient control is carried out separately in the transient state and precise control in the steady state. However, this control strategy is more complex, when the controller is applied to an induction motor. Therefore, self-tuning organizing fuzzy controllers have been developed in various fields [10].

In this paper, to remove defects of the fuzzy controller, the direct FPID controllers are designed by using the conventional PD+I controller design method [11]. To obtain the increment of fuzzy control input, this method directly applies the control gains to a PID control input concept. The increment of fuzzy control input is executed using (16).

$$du = k_p e + k_i \cdot ie + k_d \cdot de \qquad (16)$$

Where, k_i denotes integral coefficient, the values of e, de and ie are described as follows:

$$e_o \le e \le e_m \qquad (17)$$

$$de_o \le de \le de_m \qquad (18)$$

$$ie_o \le ie \le ie_m \qquad (19)$$

Where e_m, de_m and ie_m are the maximum values and e_o, de_o and ie_o represent the minimum values of error, derivative of error and integral of error, respectively. The fuzzy sets of *e*, *de* and *ie* are described as Fig. 3:

Using the above-mentioned content, each fuzzy rule can be implemented as (20) by a simplified fuzzy reasoning method.

$$\begin{array}{ll}
\text{Rule 1: } & e_o \text{ and } de_o \text{ and } ie_o \Rightarrow f_1 \\
\text{Rule 2: } & e_o \text{ and } de_o \text{ and } ie_m \Rightarrow f_2 \\
\text{Rule 3: } & e_o \text{ and } de_m \text{ and } ie_o \Rightarrow f_3 \\
\text{Rule 4: } & e_o \text{ and } de_m \text{ and } ie_m \Rightarrow f_4 \\
\text{Rule 5: } & e_m \text{ and } de_o \text{ and } ie_o \Rightarrow f_5 \\
\text{Rule 6: } & e_m \text{ and } de_o \text{ and } ie_m \Rightarrow f_6 \\
\text{Rule 7: } & e_m \text{ and } de_m \text{ and } ie_o \Rightarrow f_7 \\
\text{Rule 8: } & e_m \text{ and } de_m \text{ and } ie_m \Rightarrow f_8 \\
\text{Fact : } & e \qquad de \qquad ie
\end{array} \qquad (20)$$

Where,

$$f_1 = k_p \cdot e_0 + k_d \cdot de_0 + k_i \cdot ie_0,$$
$$f_2 = k_p \cdot e_0 + k_d \cdot de_0 + k_i \cdot ie_m,$$
$$f_3 = k_p \cdot e_0 + k_d \cdot de_m + k_i \cdot ie_0,$$
$$f_4 = k_p \cdot e_0 + k_d \cdot de_m + k_i \cdot ie_m,$$
$$f_5 = k_p \cdot e_m + k_d \cdot de_0 + k_i \cdot ie_0,$$
$$f_6 = k_p \cdot e_m + k_d \cdot de_0 + k_i \cdot ie_m,$$
$$f_7 = k_p \cdot e_m + k_d \cdot de_m + k_i \cdot ie_0,$$
$$f_8 = k_p \cdot e_m + k_d \cdot de_m + k_i \cdot ie_m.$$

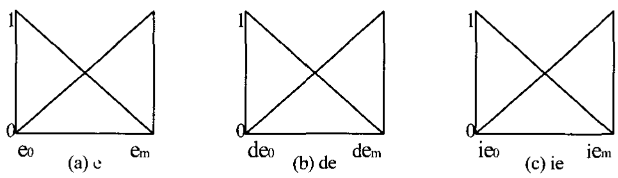


Fig. 3. Fuzzy sets of *e*, *de* and *ie*

As a result of fuzzy reasoning, the inferred output of the fuzzy controller is described as follows:

$$\begin{aligned} f = du &= \frac{f_{11} + f_{22}}{h_{11} + h_{22}} = f_{11} + f_{22} \\ &= k_p \cdot e + k_d \cdot de + k_i \cdot ie \end{aligned} \qquad (21)$$

Where

$$f_{11} = abc \cdot f_1 + ab(1-c) \cdot f_2 + a(1-b)c \cdot f_3 + a(1-b)(1-c) \cdot f_4,$$

$$f_{22} = (1-a)bc \cdot f_5 + (1-a)b(1-c) \cdot f_6 + (1-a)(1-b)c \cdot f_7 + (1-a)(1-b)(1-c) \cdot f_8,$$

$$h_{11} = abc + ab(1-c) + a(1-b)c + a(1-b)(1-c),$$

$$h_{22} = (1-a)bc + (1-a)b(1-c) + (1-a)(1-b)c + (1-a)(1-b)(1-c),$$

$$h_{11} + h_{22} = 1.$$

a , b and c mean weighting functions which are described as follows:

$$a = \mu(e_o(e)) = \frac{e_m - e}{e_m - e_o},$$

$$b = \mu(de_o(de)) = \frac{de_m - de}{de_m - de_o},$$

$$c = \mu(ie_o(ie)) = \frac{ie_m - ie}{ie_m - ie_o}.$$

The inferred output of the fuzzy controller, as seen in (21), will be used as the control input of the induction motor.

4. Auto-tuning methods of FPID controller

4.1 Heuristic algorithm

Fuzzy controllers achieved inferred values of the control inputs using triangular or bell-shaped membership functions. Recent literature has suggested that other forms of input membership function can be used to provide different properties for the controller. However, the triangular membership functions illustrated in Fig. 4 provide an adequate means for developing control capabilities and so are generally used in various applications.

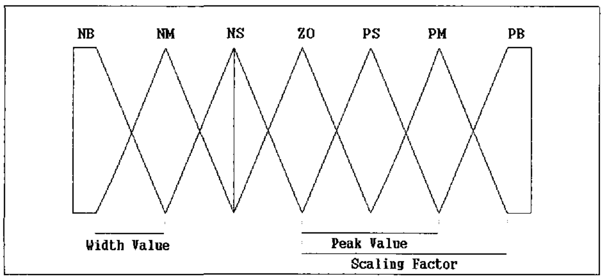


Fig. 4. Scaling factors of membership function

The changes of fuzzy control signals in the fuzzy look-up table have much influence on the performance of a system. Therefore, the use of estimated control rules obtains better results in terms of stability, robustness to parameters variation and disturbances rejection.

A fuzzy model [11] consists of a finite number of fuzzy implication rules. The fuzzy modeling relates to the construction of fuzzy rules based on a set of input reference command signals and output measurements. Using this input-output data set, fuzzy clustering method separates this data set into several local sets so that it provides an accurate representation of the system's behavior. The fuzzy clustering method, that is, a batch-mode unsupervised classification scheme, provides an analytical way for the structure of the fuzzy model.

As an example, we consider the fuzzy PD control of induction motor. The estimated look-up table of control rules is generally tuned as Table 1(a), using the heuristic algorithm. Table 1(b) shows the revised look-up table, when the shaded PM in the Table 1(a) changes to the shaded PB. Fig. 5 depicts the characteristics of induction motor speed, according to the change of a control rule. The settling time and the overshoot decreased, while the rise time increased, in Table 1(b). Therefore, according to an objective function, we can choose the PM or PB and obtain the best optimal look-up table through the iteration of the same process.

Table 1(a). Change of tuned look-up table

e \ de	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NB	NM	NS	ZO	PS
NS	NB	NB	NM	NS	ZO	PS	PM
ZO	NB	NM	NS	ZO	PS	PM	PB
PS	NM	NS	ZO	PS	PM	PB	PB
PM	NS	ZO	PS	PM	PB	PB	PB
PB	ZO	PS	PM	PB	PB	PB	PB

Table 1(b). Change of revised look-up table

e \ de	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NB	NM	NS	ZO	PS

NS	NB	NB	NM	NS	ZO	PS	PM
ZO	NB	NM	NS	ZO	PS	PM	PB
PS	NM	NS	ZO	PS	PM	PB	PB
PM	NS	ZO	PS	PM	PB	PB	PB
PB	ZO	PS	**PB**	PB	PB	PB	PB

In the paper, the defined weighted objective function appreciates to the fitness of induction motor as in (21) [13].

$$F_H(k) = s(k) + \frac{1}{2}r(k) + 10\,\frac{v_p(k) - v_{ref}(k)}{v_{ref}(k)} \quad (21)$$

Where $s(k)$ is settling time, $r(k)$ is rising time, $v_p(k)$ is maximum overshoot, and $v_{ref}(k)$ is reference speed.

The minimum value of $F_H(k)$ will reveal the optimal result. The fitness of Table 1(a) and (b) is 4.65 and 4.25, respectively. At that point, the response of Table 1(b) shows improvement over the response in Table 1(a). The heuristic method generally shows that the control system not only produces a desired response but also maintains the stability of the system. However, this method requires the expertise of an induction motor specialist.

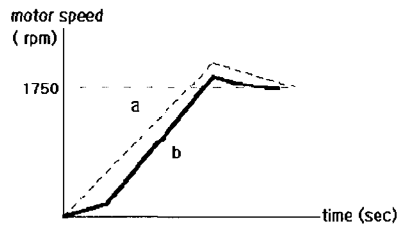


Fig. 5. Motor speed responses by change of fuzzy control rule

4.2 Genetic algorithms

Genetic algorithms (GA's) [7],[12] are directed to random search techniques, which can find the global optimal solution in complex multidimensional search spaces. GA's employ different genetic operators to manipulate individuals in a population of the solution over several generations to improve their fitness gradually. Normally, the parameters to be optimized are represented in a binary string.

To start the optimization, GA's use randomly produced initial solutions created by a random number generator. This method is preferred when prior knowledge about the problem is not available.

Fig. 6 shows the general flow chart of GA's [7]. Three genetic operators are used in order to generate and explore the neighborhood of a population and to select a new generation. These operators are reproduction, crossover and mutation. After randomly generating the initial population of N solutions, the GA's use the three genetic operators to yield N new solutions per iteration operation. In the selection operation, each solution of the current population is evaluated by its fitness value obtained from an objective function. Individuals with higher fitness value are selected for survival in the next generation.

For the purposes of this paper, to conduct the crossover operator, input variables are multiplied by 1000, rounded off by fractions, transformed into integers and converted to a binary digital system. These integers become new input variables. Takagi's formula [13] is used as the objective function that is defined by the function of input variables as seen in (22).

$$F_G(k) = \sqrt{\left(e(k)^2 + de(k)^2 + ie(k)^2\right)} \quad (22)$$

Where $e(k)$, $de(k)$ and $ie(k)$ represent error, derivative of error and integral of error show input variables, respectively.

The population numbers 10 and the number of chromosomes gains 20 through the conversion of five decimal places to a binary system. The selected genetics use probability theory and random variables. The crossover and mutation rates also use random variables. The algorithm is repeated until a predefined result has been produced.

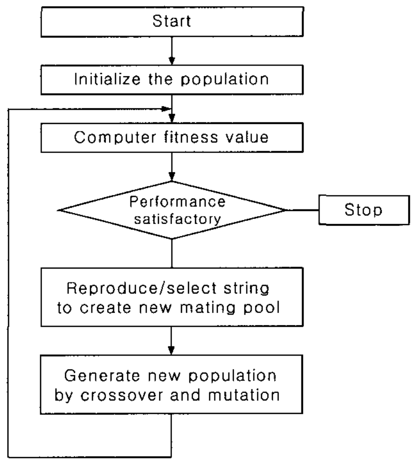


Fig. 6. Flow chart using GA's

Through the genetic algorithm, a look-up table records the optimized results off-line and is used to reference the on-line system. After each scope of input variables, $e(k)$, $de(k)$ and $ie(k)$ are divided into 5, 30 or 100 input partitions, the optimal fitness is calculated by the fuzzy reasoning and the genetic algorithm at each partition. Each table consists of 5^3 =125, 30^3 =27,000 or 100^3=1,000,000 databases by the optimal fitness of each partition. If each partition is divided into equal intervals, the response of the steady state produces the worst result. Therefore, each partition is divided by its linear proportion to the square of error, referred to as steady state.

The scope of $e(k)$ is partitioned to set the difference between reference speed and initial speed of -100% and 100% to an overshoot of 100%. The scope of $de(k)$ is partitioned from 1000 at 100% and -1000 at -100%, because $de(k)$ approaches infinity. The scope of $ie(k)$ is partitioned at 1 to 100% and -1 to -100%, because $ie(k)$ is limited from -1 to 1.

As the number of partitions increases, the best results can be obtained. However, increasing partitions could hinder the performance due to limited computer capability and the low access speed. Therefore it is suitable to select partitions between 10 and 100.

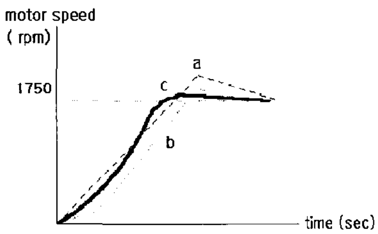


Fig. 7. Motor speed response of optimized fuzzy control rule by GAs.

The genetically optimized pseudo-on-line method, that is, hybrid GFPID, uses the estimated look-up table based on the genetic algorithm. This method improves performance of the on-line or off-line GA's. Heuristic algorithms that change fuzzy control rule in the experiment perform better in the steady-state error, but not in the transient-state error. The genetically optimized pseudo-on-line method provides high results in the steady–state and transient-state errors. The **c** line in Fig. 7 represents the motor speed response based on (22) when the hybrid GFPID is applied, using an optimized fuzzy control rule by GA's.

Optimized fitness results in the case of 5 input partitions for each input, is arranged in Table 2. One partition of each input, $de(k)$, $ie(k)$ and $e(k)$ respectively, represent one fifth of 200%, that is, 40%. The increase of partitions increases the complexity of the table. By executing the hybrid GFPID control strategy on the off-line procedure, the optimal fitness is calculated by the fuzzy reasoning and the genetic algorithm at each partition. Table 2 shows the look-up table that consists of 5^3=125 databases for optimal fitness measurement, using 5-input partitions. The look-up table is applied to on-line control of the induction motor. Similarly, using 30 or 100 input partitions, a look-up table can also consist of 30^3 = 27,000 or 100^3=1,000,000 databases, and the look-up table also is applied to on-line control of the induction motor.

Table 2. Optimized results of each fitness measurement using 5 partitions

ie(k)	*e(k)* \ *de(k)*	Partition1	Partition2	Partition3	Partition4	Partition5
Partition1	Partition1	0.0018	0.0037	0.0024	0.0104	0.0028
	Partition2	0.0043	0.0063	0.0052	0.0098	0.0050
	Partition3	0.0036	0.0044	0.0039	0.0073	0.0049
	Partition4	0.0078	0.0094	0.0081	0.0135	0.0095
	Partition5	0.0040	0.0064	0.0049	0.0080	0.0078
Partition2	Partition1	0.0019	0.0037	0.0023	0.0094	0.0028
	Partition2	0.0044	0.0064	0.0052	0.0098	0.0051
	Partition3	0.0037	0.0050	0.0039	0.0084	0.0049
	Partition4	0.0078	0.0095	0.0081	0.0138	0.0095
	Partition5	0.0040	0.0065	0.0050	0.0082	0.0030
Partition3	Partition1	0.0020	0.0038	0.0023	0.0094	0.0029
	Partition2	0.0044	0.0064	0.0052	0.0085	0.0050
	Partition3	0.0038	0.0051	0.0040	0.0085	0.0050
	Partition4	0.0079	0.0095	0.0082	0.0138	0.0095
	Partition5	0.0042	0.0066	0.0051	0.0083	0.0031
Partition4	Partition1	0.0019	0.0037	0.0024	0.0092	0.0028
	Partition2	0.0043	0.0063	0.0051	0.0086	0.0049
	Partition3	0.0038	0.0061	0.0041	0.0084	0.0049
	Partition4	0.0078	0.0094	0.0094	0.0135	0.0096
	Partition5	0.0040	0.0064	0.0051	0.0082	0.0030
Partition5	Partition1	0.0018	0.0036	0.0023	0.0091	0.026
	Partition2	0.0042	0.0063	0.0050	0.0085	0.0048
	Partition3	0.0037	0.0059	0.0041	0.0082	0.0049
	Partition4	0.0077	0.0096	0.0093	0.0130	0.0029
	Partition5	0.0038	0.0062	0.0050	0.0081	0.0025

5. Simulations and Results

In this section, several simulations have examined the feasibility of the proposed pseudo-on-line algorithm for the induction motor system.

First, we consider the type of fifth-order nonlinear differential equations in (23) that are generally treated as the accepted model for simulations of induction motors.

$$\begin{aligned}\frac{d}{dt}i_{qs} &= \frac{-1}{L_sL_r - L_m^2}\left(R_sL_ri_{qs} + \omega_rL_m^2i_{ds} - R_rL_mi_{qr} + \omega_rL_rL_mi_{dr} - L_rv\right)\\ \frac{d}{dt}i_{ds} &= \frac{-1}{L_sL_r - L_m^2}\left(-\omega_rL_m^2i_{qs} + R_sL_ri_{ds} - \omega_rL_rL_mi_{qr} + R_rL_mi_{dr} - L_rv_{ds}\right)\\ \frac{d}{dt}i_{qr} &= \frac{-1}{L_sL_r - L_m^2}\left(-R_sL_mi_{qs} - \omega_rL_sL_mi_{ds} + R_rL_si_{qr} - \omega_rL_rL_si_{dr} + L_rv_{qs}\right)\\ \frac{d}{dt}i_{dr} &= \frac{-1}{L_sL_r - L_m^2}\left(\omega_rL_sL_mi_{qs} - R_sL_mi_{ds} + \omega_rL_rL_si_{qr} + R_rL_si_{dr} + L_rv_{ds}\right)\end{aligned} \quad (23)$$

Also, consider the mechanical torque equation in (24) [8].

$$T_m = J\left(\frac{2}{P}\right)\frac{d}{dt}\omega_r + B\left(\frac{2}{P}\right)\omega_r + T_l \quad (24)$$

Where J denotes the moments of inertia, 0.0179Kg m^2, $\boldsymbol{B}$ the frictional coefficient of load and motor, 1.464Kg m^2 and T_l the load torque, 10Kg, respectively. In the induction motor, as the electromagnetic torque in (11) is equal to the mechanical torque in (24), the relationship obtained is seen in (25).

$$\frac{d}{dt}\omega_r = \frac{3L_m}{2J}\left(i_{qs}i_{dr} - i_{ds}i_{qr}\right) - \frac{B}{J}\omega_r - \frac{1}{J}T_l \quad (25)$$

Solving (23) and (25) by the Runge-Kutta method, the numerical solution was obtained

Table 3. Motor parameters

Normal output	P_0	1.1*Kw*
Normal rotational frequency	*RPM*	1000*rpm*
Stator resistance	R_s	0.2842 Ω
Rotor resistance	R_r	0.2878Ω
Stator leakage inductance	L_{ls}	0.02827Ω
Rotor leakage inductance	L_{lr}	0.02827Ω
Magnetizing inductance	L_m	0.02682Ω
Leakage coefficients	σ	0.116
Number of pole pairs	*P*	3

Table 3 shows the rated values and the nominal parameters of a tested induction machine, 1.1 Kw.

Simulation results are depicted in Fig. 8 ~ 13, when the motor speed is steeply changed from -500[rpm] to 500[rpm] and the GFPID control technique proposed in the previous section is applied to a tested induction motor. Simulation outputs are compared with the FPID and PID. In these figures, a case of 30 levels is compared in the viewpoint of motor speed, torque component current and emulator output. We use (27) as a performance index (PI) of the induction motor. Here, e is defined as speed error of the induction motor.

$$PI = \int\sqrt{e^2} \quad (27)$$

Where e is defined as a speed error of induction motor.

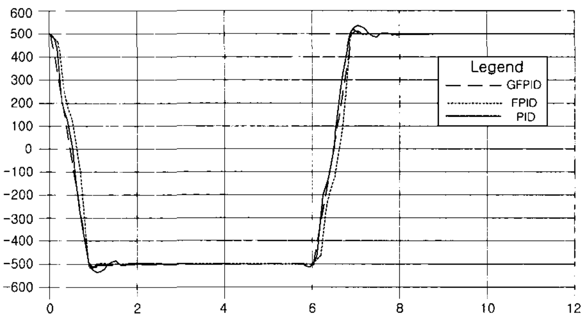


Fig. 8. Motor speed (x-sec., y-rpm, no load)

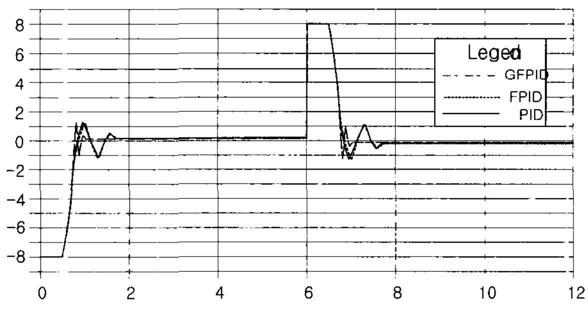


Fig. 9. Torque component current (x-sec., y-mA, no load)

As shown in Fig. 8~10 with no load, the proposed controller produces high performances in areas of motor speed, torque component current and emulator output, in specific electrical and mechanical time constants. Namely, the GFPID reduces the settling time and improves maximum overshoot, but has almost no changes in rise time and the performance index, over the FPID and PID. We can further see that the torque component current has little non-linearity with oscillation, because it must generate the same emulator output as motor speed.

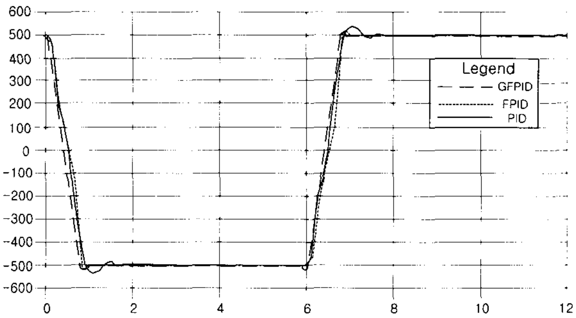


Fig. 10. Emulator output (x-sec., y-rpm, no load)

As shown in Fig. 11~13 with load, the proposed controller shows almost the same performance for the rise time and the performance index as the FPID and PID. However, the GFPID improves maximum overshoot and reduces the settling time. The torque component current of the GFPID has a complex non-linearity; that is, it oscillates noticeably in the steady state. This is a general characteristic of a powerful controller like the ones in FPID and GFPID, where small errors occur in the large reactions. Increasing amplitude of oscillation could have difficulty in realizing the torque component current.

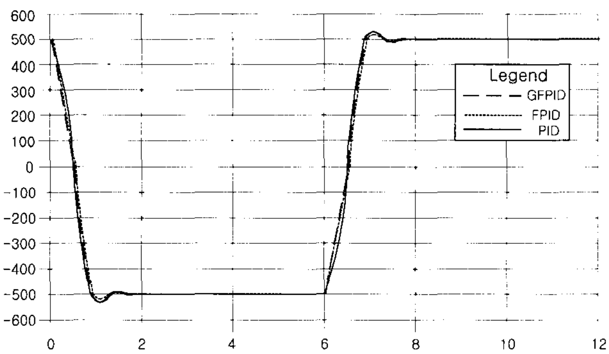


Fig. 11. Motor speed (x-sec., y-rpm, load)

Table 4 compares abilities of each controller under no additional load. Table 5 compares abilities of each controller under additional load. It can be seen from Tables 4 and 5 that the GFPID control strategy gives very good results relative to maximum overshoot and settling time, but only slight improvement in rise time.

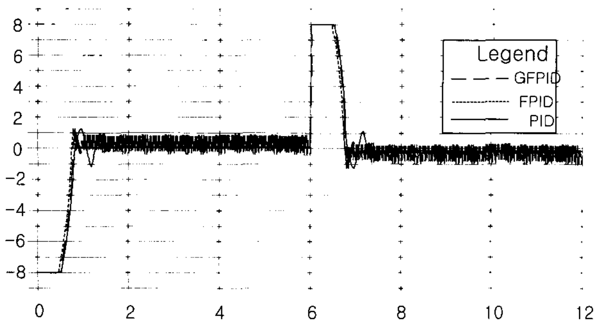


Fig. 12. Torque component current (x-sec., y-mA, load)

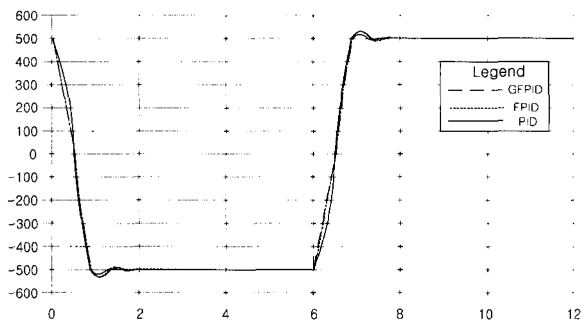


Fig. 13. Emulator output (x-sec., y-rpm, load)

The explanation for the GFPID control strategy results from a significant improvement in maximum overshoot and settling time, but a slight improvement in rise time. This is due to the look-up table that consists of estimated parameters with optimal damping factors and damping constants is use. If we choose different factors or constants, this strategy results in significant improvements in rise time, but not in maximum overshoot and settling time.

The improvements in maximum overshoot and settling time, using the GFPID control strategy, lead to observable improvements in the running of an induction motor. For example, when using a GFPID controller, it has been observed that the optimized parameters properly control the induction motor.

By optimally estimating the off-line parameters of the FPID controller using the genetic algorithm, making the optimized look-up table using the estimated parameters and driving in the on-line induction motor, the speed control of the induction motor shows significant improvement for no load and load.

Table 4. Ability of motor speed for each controller with no load

Method \ Evaluation		Estimated value (PI)	Rise time(s)	Maximum overshoot(%)	Settling time(s)
PID		13.577	0.640	4.4	1.844
FPID		12.425	0.639	4.3	1.728
GFPID	5 partitions	12.017	0.639	2.7	1.710
	30 partitions	11.532	0.634	1.9	1.324
	100 partitions	10.044	0.621	0.1	1.013

Table 5. Ability of motor speed for each controller with load

Method \ Evaluation		Estimated value (PI)	Rise time(s)	Maximum overshoot(%)	Settling time(s)
PID		13.834	0.641	4.5	3.557
FPID		12.664	0.639	4.4	1.794
GFPID	5 partitions	12.138	0.640	2.9	1.720
	30 partitions	11.624	0.635	1.9	1.328
	100 partitions	10.046	0.621	0.1	1.013

6. Conclusions

This paper proposed a hybrid induction motor control method using a genetically optimized pseudo-on-line method. Optimization resulted from the use of a look-up table based on genetic algorithms that do not use the gradient. However, we can achieve the global optimum of the un-constraint optimization problem. The approach to the induction motor control includes a pseudo-on-line procedure that optimally estimates parameters of a fuzzy PID (FPID) controller. This technique includes the procedures that optimally estimate the off-line parameters of the FPID controller using the genetic algorithm. The optimized look-up table uses the estimated parameters and control in on-line systems with non-linearity at real-time. To prove the higher performance, the proposed hybrid genetic fuzzy PID (GFPID) controller was applied to speed control of a 3-phase induction motor and its computer simulation was carried out.

The simulation results of the GFPID controller are as follows:

1. The speed control of induction motor with non-linearity performs better than the conventional FPID and PID on load or no load.
2. Dividing the input region into partitions and making an optimized look-up table based on genetic algorithm, the GFPID made on-line control with off-line performance at real-time possible.
3. The GFPID control results showed significant improvement in maximum overshoot and settling time at a transient state, especially, when a number of partitions of the input variable are increased.
4. The performance improvements without load or with load could result in improved operations of induction motors in a number of applications such as robotic systems, medical procedures and other uses.
5. The GFPID controller could make it possible to control the speed of electric vehicles in drive systems of induction motors.

The contribution of this paper is the introduction of a high performance hybrid form of induction motor control that makes on-line and real-time control of the drive system possible. The possibility of optimizing the control system with complex non-linearity will lead to designing the universal controller based on a genetic algorithm.

Acknowledgements

This work was supported by a research fund provided by Won kwang Health Science College in 2002.

References

[1] B. K. Bose, "Power Electronics and AC Drivers", Prentice-Hall Book, 1986.

[2] J. Cleland and W. Turner, "Fuzzy logic control of AC induction motor", IEEE Research Tringle Institute, pp. 843 – 850, 1992.

[3] P. Vas, "Vector Control of AC Machines" Clarendon Press. Oxford. 1990.

[4] Y. Yoshinari, W. Pedrycz and K. Hirota, "Construction of fuzzy models through clustering techniques", Fuzzy sets Syst., vol. 54, no. 2. pp. 157-165, 1993

[5] M. Mizumoto, "Realization of PID control by fuzzy control method", IEEE Division of Information and Computer Sciences, pp. 709-715, 1992.

[6] W. C. Dangherity, B. Rathakrishnan and J. Ten, "Performance evaluation of a self-tuning fuzzy controller", IEEE Department of Computer Science Texas A & M University College Station TX 77843, pp. 389-397, 1992.

[7] R. Caponetto, L. Fortuna, S. Graziani and M. Xibilia, "Genetic algorithms and applications in system engineering: a Survey", Trans. Inst. Meas. Control, vol. 15, pp. 143-156, 1993.

[8] P. C. Krause, "Analysis of Electric Machinery", MeGraw-Hill Book, 1987.

[9] Y. F. Li and C. C. Lau, "Development of Fuzzy Algorithms for Servo Systems", IEEE Control System Magazine, Apr., pp.65-71, 1989.

[10] L. Zheng, "A practical guide to tune of proportional and integral (PI) Like fuzzy controller", IEEE Yamatake-Honey Co. Report, Japan, 1992.

[11] H. A. Malki, H. Li and G. Chen, "New design and stability analysis of fuzzy proportional-derivative control system" IEEE Transactions on Fuzzy Systems, vol. 1, no. 6, pp. 245-254, November 1994.

[12] J. H. Holland, "Adaptation in Neural and Artificial Systems", Ann Arbor, MI: University of Michigan Press Book, 1975

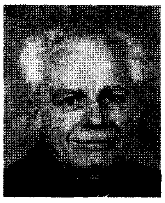
[13] T. Takagi and M. Sugeno, "Fuzzy identification of system and its applications to modeling and control", IEEE Transactions on SMC-15, no.1 Jan./Feb., pp.116-132, 1985.

[14] S. H. Chae, "An induction motor drive using an auto-tuning fuzzy PID control algorithm", Chung-Ang University, Master's thesis, 1992.

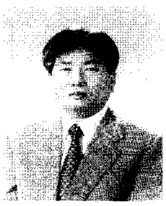
[15] M. Moallem, B. Mirzarian, O. A. Mohammed and C. Lucas, "Multi-objective genetic-fuzzy optimal design of PI controller in the indirect field oriented control of an induction motor", IEEE Transactions on Magnetics, vol. 37, no.5, pp. 3608-3612, September, 2001.

[16] B. Hu, G. K. I. Mann and R. G. Gosine, "New methodology for analytical and optimal design of fuzzy PID controller", IEEE Transactions on Fuzzy Systems, vol. 7, no. 5, October, pp. 521-539, 1999.

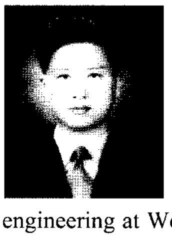
Tae-chon Ahn was born in Incheon, Korea on October 11, 1955. He received B.S., M.S. and Ph.D. degrees in Electrical Engineering from Yonsei University, Korea in 1978, 1980 and 1986, respectively. From 1980 to 1981, he was employed as a researcher at Korea Power Co., Korea. From 1987 to 1988, he was a visiting scholar in the department of Control System Engineering at Upscale University, Sweden. From 1996 to 1997, he was a visiting professor in the department of Electrical and Computer Engineering at Georgia Institute of Technology, USA. Since 1981, he has been with the School of Electrical and Electronics Engineering at Won kwang University, Korea, where he is currently a Professor. He is engaged in research on digital granular control, real-time data measuring and processing, FPGA design, intelligent systems, stability and control of synchronous machines.



James F. Peters received B.S. and M.S. degrees in Mathematics from the University of California, and a Ph.D. in Computer Science from the University of Kansas, USA. He is currently an Associate Professor in the Dept. of Electrical and Computer Engineering at the University of Manitoba, Canada. His current research includes Computational intelligence, identification and classification, rough sets, rough neural networks, rough control, rough and fuzzy measures, rough and fuzzy description in software engineering, and modeling, design and analysis of intelligent systems.



Jong-seok Lee was born in Iksan, Korea, in 1957. He received B.S. and M.S. degrees in Electrical Engineering from Won kwang University, Iksan, Korea, in 1984 and 1988, respectively, and a Ph.D. in Electric Engineering from Won kwang University, Iksan, Korea, in 2002. Since 1996, he has been with the Department of Radiotechnology, College of Won kwang Health Science, Iksan, Chonbuk, Korea, where he is currently a Professor.



Kyung-Won Jang was born in Korea in 1974. He received B.S. and M.S. degrees in control and instrument engineering from Won kwang University, Iksan, Korea, in 2001 and 2003, respectively. He is currently working towards a Ph. D. in control and instrumentation engineering at Won kwang University, Iksan, Korea. His current research deals with the model based controller design, pattern recognition, granular computing and its applications.