Intelligent Parameter Estimation of a Induction Motor Using Immune Algorithm

Dong Hwa Kim, Jae Hoon Cho
Dept. of Instrumentation and Control Eng., Hanbat National University,
16-1 San Duckmyong-Dong Yuseong-Gu, Daejon City, Korea, 305-719.
E-mail: kimdh@hanbat.ac.kr, Hompage: ial.hanbat.ac.kr
Tel: +82-42-821-1170, Fax: +82-821-1164

Abstract—**This paper suggests the techniques in determining the values of the steady-state equivalent circuit parameters of a three-phase squirrel-cage induction machine using immune algorithm. The parameter estimation procedure is based on the steady state phase current versus slip and input power versus slip characteristics. The proposed estimation algorithm is of a nonlinear kind based on clonal selection in immune algorithm. The machine parameters are obtained as the solution of a minimization of least-squares cost function by immune algorithm. Simulation shows better results than the conventional approaches.-**

I. Introduction

In an ac induction motor drive, the electrical parameters are, generally, determined via the classical analysis and no-load tests. Estimation of the performance behavior of an induction machine is also done by plotting the steady-state slip curves. Generally to obtain the parameter, one must use the equivalent circuit relations and the experimental results obtained from the above-mentioned classical analysis. Therefore, the parameter values obtained by direct classical approaches or experimentation can reveal significant differences in the entire range of slip varying from 0 to 1. To describe the performance of the induction machine more precisely and to reduce the differences between the estimated and real performances, one must modify the parameters obtained from the classical analysis. To achieve this purpose in motor, the use of system identification algorithms based on the artificial algorithm appears to be a very promising approach. These algorithms allow one to take into account the effect of measurement errors, disturbances, and random signals on the estimated parameters. Since the equations relating the phase current to the slip and the circuit parameters involve many variables and are nonlinear, parameters can have difference values in case of the change of load.

This fact does not enable one to directly use the many parameter estimation procedures existing in the literature.

In this paper, obtaining optimal parameters of the equivalent circuit of a squirrel-cage induction machine is suggested by immune algorithm. During the execution of the estimation algorithm, we use the steady-state characteristic curves of both the input power and the stator current to adjust the initial parameter vector. The difference between the proposed immune based optimal parameter and the classical procedure is compared.

II. Induction Motor Model

A squirrel-cage induction machine supplied with a three-phase symmetrical voltage source can be described using the equivalent circuit shown in Fig. 1.

In case the stator current the input power, the equations of and the electromagnetic torque for a squirrel induction motor can be deduced from the circuit of Fig. 1 and are expressed as follows:

$$I(s) = V\sqrt{\frac{C^2 + D^2}{A^2 + B^2}}$$

$$P(s) = 3V^2 \frac{AC - BD}{A^2 + B^2}$$

$$T(s) = 3V^2 \frac{p}{\omega} \frac{R_{fe}^2 R_r / s}{A^2}$$

$$A = R_s\left(1 + \frac{X_r}{X_m}\right) + \left(1 + \frac{X_s}{X_m}\right)\frac{R_s}{S}$$

$$B = X_r + X_s\left(1 + \frac{X_r}{X_m}\right) - R_s\left(R_r / S \frac{R_r / S}{X_m}\right) \quad (1)$$

$$C = 1 + \frac{X_r}{X_m}$$

$$D = \frac{R_r / S}{X_m}$$

$$R_r = \frac{R_{r1}R_{r2}(R_{r1} + R_{r2}) + (R_{r1}X_{r2}^2 + R_{r2}X_{r1}^2)S^2}{(R_{r1} + R_{r2})^2 + (X_{r1} + X_{r2})^2 S^2}$$

$$R_r = \frac{X_{r1}X_{r2}(X_{r1} + X_{r2})S^2 + R_{r1}^2 X_{r2} + R_{r2}^2 X_{r1}}{(R_{r1} + R_{r2})^2 + (X_{r1} + X_{r2})^2 S^2}$$

$$I(s) = V\sqrt{\frac{C^2 + D^2}{A^2 + B^2}}$$

$$P(s) = 3V^2 \frac{AC - BD}{A^2 + B^2}$$

$$\theta = [\,R_{r1} \quad R_{r2} \quad X_{r1} \quad X_{r2} \quad X_s \quad R_s \quad X_m\,]^T$$

In the above equations, R_s , R_r , and R_{fe} are the stator, rotor, and iron losses, respectively. Also, X_s, X_r, *and* X_m are the stator leakage reactance, rotor leakage reactance, and magnetizing reactance. For neglecting the iron losses of a double-cage motor, one must add a second branch in parallel with the magnetizing reactance.

III. Optimal Parameter Estimation for Induction Motor using Immune Algorithm

A. Motor Parameter for Optimal Parameter Selection

In order to determine model parameters from the slip curves of the equivalent circuit, reference [1] uses a nonlinear curve-fitting problem stated as the solution of the following minimization problem:

$$\min_{\theta\in\Omega} = J(\theta) = \frac{1}{N}\sum_{i=1}^{N}[y_i - y(s_i, \theta)]^2 \qquad (2)$$

where $J(\theta)$ is least squares cost function obtained by the sum of the squares of the differences between the experimental and calculated slip curves, Ω is parameter space depending on the number of parameters to be estimated, y_i is the experimental data value collected from machine, $y(s_i, \theta)^2$ is nonlinear function relating the measured data, the circuit parameters, and the slip, and θ is parameter vector pertaining to Ω. Therefore, in case of double cage, dimension of parameter vector Ω is defined as:

$$\theta = [R_{ri},\ R_{r2},\ X_{r1},\ X_{r2},\ X_s,\ R_s,\ X_m]^T. \qquad (3)$$

The above mentioned specific equation is for depends on the kind of available experimental data and for obtaining a parameter vector that minimizes the quadratic performance index defined by equation (2). In this case, since one must deal with a nonlinear algorithm to acquire the desired solution, some numerical problems may arise or a direct approach would require writing down the normal equations for solving them. The methods for numerical minimization of performance index (2) might be modified to update the estimated parameter vector according to load change.

B. Clonal Selection Algorithm for Optimal Parameter Selection

In this paper, clonal selection algorithm as depicted in Fig. 2 is introduced. That is, when an antibody on the surface of a B cell binds an antigen, that B cell becomes stimulated. The level of stimulation depends not only on how well the B cell's antibody matches the antigen, but also how it matches other B cells in the immune network [3-4]. The stimulation level of the B cell also depends on its affinity with other B cells in the immune network. This network is formed by B cells possessing an affinity to other B cells in the system. If the stimulation level rises above a given threshold, the B cell becomes enlarged and if the stimulation level falls below a given threshold, the B cell die off. The more neighbors a B cell has an affinity with, the more stimulation it will receive from the network, and vice versa. Against the antigen, the level to which a B cell is stimulated relates partly to how well its antibody binds the antigen. We take into account both the strength of the match between the antibody and the antigen and the B cell object's affinity to the other B cells as well as its enmity. Therefore, generally the concentration of i-*th* antibody, which is denoted by δ_i, is calculated as follows [21]:

$$\frac{dS_i(t)}{dt} = \left(\begin{array}{l}\alpha\sum_{j=1}^{N} m_{ji}\delta_j(t) \\ -\alpha\sum_{k=1}^{N} m_{ik}\delta_k(t) + \beta m_i - \gamma_i\end{array}\right)\delta_i(t), \qquad (3a)$$

$$\frac{d\delta_i(t)}{dt} = \frac{1}{1+\exp\left(0.5 - \frac{dS_i(t)}{dt}\right)}, \qquad (3b)$$

In Eq. (3), N is the number of antibodies, and α and β are positive constants. m_{ji} denotes affinities between antibody j and antibody i (i.e. the degree of interaction), m_i represents affinities between the detected antigens and antibody i, respectively.

B. Computation Procedure For Optimal Selection

The coding of an antibody in an immune network is very important because a well designed antibody coding can increase the efficiency of the controller. As shown in Fig. 3, there are seven antibodies for parameters, $\theta = [R_{ri},\ R_{r2},\ X_{r1},\ X_{r2},\ X_s,\ R_s,\ X_m]^T$ and object function for equation (2). Each parameter is specified in memory cell of immune network.

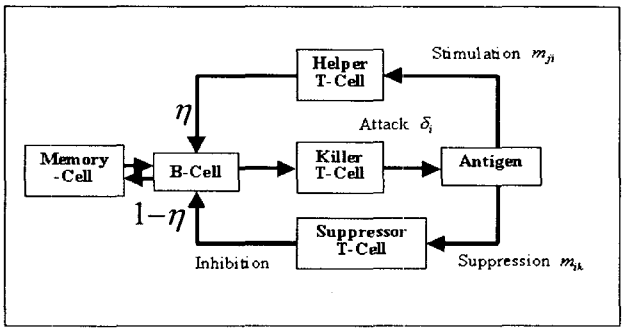

Fig. 1. Dynamic relationship between cells, antigen.

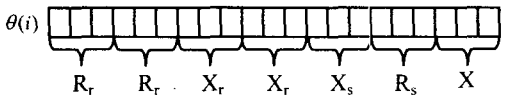

Fig. 2. Allocation structure of each parameter.

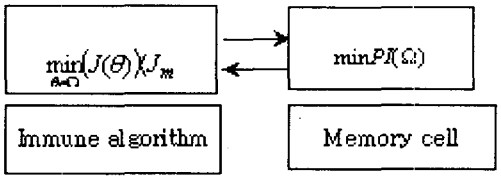

Fig. 3 Immune algorithm based computational structure for optimal parameter selection

[Step 1] Initialization and recognition of antigen: The immune system recognizes the invasion of an antigen, which corresponds to parameter for the optimization problem as shown in Fig. 3.

[Step 2] Product of antibody from memory cell: The immune system produces the antibodies that were effective to kill the antigen in the past. This is implemented by recalling a past successful solution from memory cell.

[Step 3] Antibody with the best fitness value obtained by calculation for searching an optimal solution is stored in memory cell.

[Step 4] Differentiation of lymphocyte: The B-lymphocyte cell, the antibody that matched the antigen, is dispersed to the memory cells in order to respond to the next invasion quickly. That is, select individuals using tournament selection and apply genetic operators (crossover and mutation) to the individuals of network.

[Step 5] Stimulation and suppression of antibody: The expected value η_k of the stimulation of the antibody is given by

$$\eta_k = \frac{m_{\varphi k}}{\sigma_k} \tag{4}$$

where σ_k is the concentration of the antibodies. The concentration is calculated by affinity. So, σ_k is represented by

$$\sigma_k = \frac{sum\ of\ antibodies\ with\ same\ affinity\ as\ m_{\varphi k}}{sum\ of\ antibodies}. \tag{5}$$

Using equation (5), an immune system can control the concentration and the variety of antibodies in the lymphocyte population. If antibody obtains a higher affinity against an antigen, the antibody stimulates. However, an excessive higher concentration of an antibody is suppressed. Through this function, an immune system can maintain the diversity of searching directions and a local minimum.

[Step 6] Calculate fitness value between antibody and antigen. This procedure can generate a diversity of antibodies by a genetic reproduction operator such as mutation or crossover. These genetic operators are expected to be more efficient than the generation of antibodies. Table 1 shows the results obtained by clonal selection.

[Step 7] If the maximum number of generations of memory cell is reached, stop and return the fitness of the best individual fitness value to network; otherwise, go to step 3.

III. SIMULATION

The clonal selection algorithm suggested in this paper is simulated and compared with genetic algorithm, recursive algorithm cited in reference [1]. In reference [1], Object function $J_1(\theta)$ is used but object function $J_2(\theta)$ is introduced for more optimal parameter selection as follows:

$$J_1(\theta) = \frac{1}{N}\sum_{i=1}^{N}[I(si) - I(si,\theta)]^2 + \frac{1}{N}\sum_{i=1}^{N}[P(si) - P(si,\theta)]^2 \tag{6}$$

$$J_2(\theta) = \frac{1}{N}\sum_{i=1}^{N}[S(i)I(si) - S(i)I(si,\theta)]^2 + \frac{1}{N}\sum_{i=1}^{N}[S(i)P(si) - S(i)P(si,\theta)]^2 \tag{7}$$

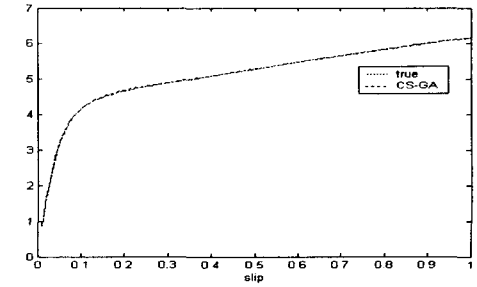

Fig. 5. Variation of I(s) by Clonal selection and true values.

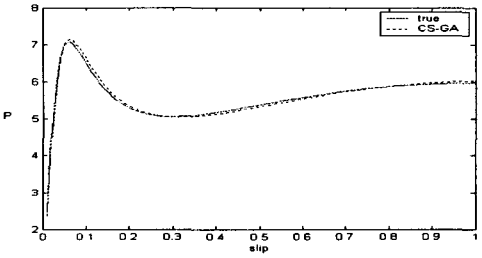

Fig. 6. Variation of P(s) by Clonal selection and true values.

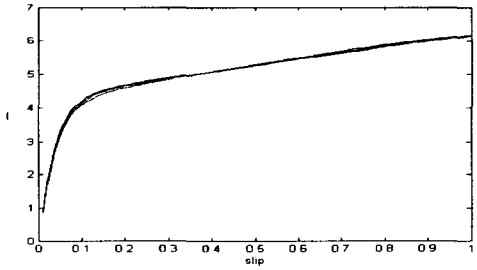

Fig. 7. Variation of P(s) by Clonal selection, GA, recursive, and true values

Fig. 5 represents variation of current curve I(s) by Clonal selection and true values and Fig. 6 is the curve of power P. Fig. 7 is variation of P(s) obtained by Clonal selection, GA, recursive, and true values and Fig. 8 shows variation of P(s) by Clonal selection (CS-GA), GA, recursive, and true values. These figures are showing that the suggested algorithm can be used in parameter estimation.

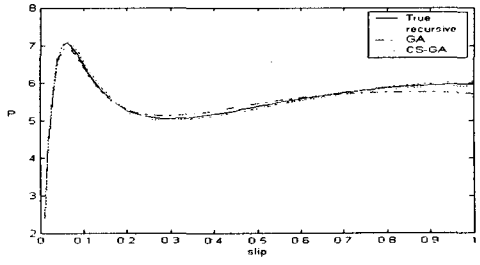

Fig. 8. Variation of P(s) by Clonal selection, GA, recursive, and true values

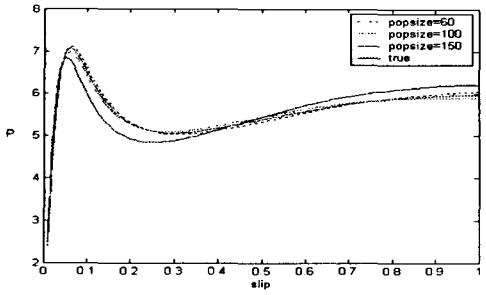

Fig. 9. Parameter variation to popsize in objective function $J_1(\theta)$.

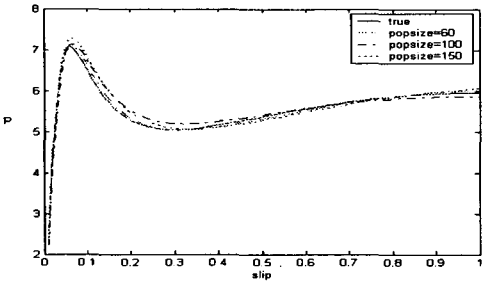

Fig. 10. Parameter variation to popsize in objective function $J_2(\theta)$.

Acknowledgments: This research is supported by the EESRI.

REFERENCES

[1] A.M.N. Lima, C.B. Jacobina, E.B. Souza Filho, " Nonlinear parameter estimation of steady-state induction machine models," IEEE Trans. Industrial Electronics, vol. 44, no. 3, 1997.

[2] J.Hickiewicz, K. Macek-Kaminska, and P.Wach, "Algorithmic methods of induction machines parameters estimation from measured slip curves," Arch. Elektrotech., vol. 1, no.72, pp. 2.

[3] Kazuyuki Mori and Makoto Tsukiyama, "Immune algorithm with searching diversity and its application to resource allocation problem," Trans. JIEE, Vol. 113 - C, No. 10, 1993.

[4] Dong Hwa Kim, "Auto-tuning of reference model based PID controller using immune algorithm," IEEE international conference on evolutionary computation, Hawaii, May 12 - 17, 2002.

[5] Dong Hwa Kim, "Intelligent tuning of a PID controller using an immune algorithm," Trans. KIEE, Vol. 51-D, No.1, pp. 2002.

[6] Dong Hwa Kim, "PID Controller Tuning of a Boiler Control System Using Immune Algorithm Typed Neural Network," ICCS2004, Poland, June 2004.

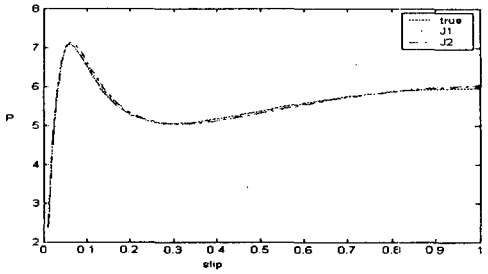

Fig. 11. Comparison of two object function ($J_1(\theta), J_2(\theta)$).

Tabel.1 Initail boundary and ture values.

gen	R_{r1}	R_{r2}	X_{r1}	X_{r2}	X_s	R_s	X_m
θ_{ture}	0.0693	0.0132	0.00843	0.1162	0.123	0.00778	4.3
xlb	0.06	0.01	0.007	0.10	0.10	0.006	4
xub	0.08	0.015	0.01	0.13	0.13	0.008	4.5

Table.2 parameter

	R_{r1}	R_{r2}	X_{r1}	X_{r2}	X_s	R_s	X_m
Recursive	0.078	0.0129	0.0164	0.121	0.1167	0.0073	4.29
GA	0.063	0.0138	0.010	0.122	0.1260	0.0074	4.21
CS-GA	0.0755	0.0135	0.009	0.1168	0.1195	0.0064	4.34

Table.3 Parameter of each object function

Obj_func	popsize	R_{r1}	R_{r2}	X_{r1}	X_{r2}	X_s	R_s	X_m
$J_1(\theta)$	60	0.0755	0.0135	0.0089	0.1168	0.1195	0.0064	4.34
	100	0.0647	0.0132	0.0081	0.1186	0.1253	0.0078	4.32
	150	0.0699	0.0115	0.0074	0.1231	0.1207	0.0066	4.42
$J_2(\theta)$	60	0.0755	0.0135	0.0091	0.1159	0.1196	0.0068	4.28
	100	0.068	0.0149	0.0081	0.1174	0.1242	0.0069	4.47
	150	0.0787	0.0136	0.0082	0.1105	0.1188	0.0077	4.30

Tabel.4 Parameter for genetion variation of the each object function

Obj_func	gen	R_{r1}	R_{r2}	X_{r1}	X_{r2}	X_s	R_s	X_m
$J_1(\theta)$	100	0.075594	0.013538	0.0089178	0.11685	0.11951	0.0064439	4.3419
	150	0.072702	0.01227	0.007611	0.11916	0.1202	0.0069193	4.1168
	200	0.072702	0.01227	0.007611	0.11916	0.1202	0.0069193	4.1168
	300	0.072702	0.01227	0.007611	0.11916	0.1202	0.0069193	4.1168
$J_2(\theta)$	100	0.075521	0.013558	0.0091887	0.11596	0.11962	0.0068187	4.2845
	150	0.075521	0.013558	0.0091887	0.11596	0.11962	0.0068187	4.2845
	200	0.075495	0.013516	0.0092333	0.11596	0.11961	0.0068168	4.4295
	300	0.0755	0.01356	0.0092373	0.11596	0.11963	0.0068197	4.3595
	400	0.0755	0.01356	0.0092379	0.11596	0.11963	0.0068197	4.3604