적응진화 알고리즘을 사용한 DC 모터 퍼지 제어기 설계에 관한 연구

김동완* · 황기현** · 이재현***

Design of a Fuzzy Logic Controller Using an Adaptive Evolutionary Algorithm for DC Series Motors

Dong-Wan Kim* · Gi-Hyun Hwang** · Jae-Hyun Lee***

요 약

본 논문에서는 적응진화알고리즘을 사용한 퍼지제어기의 설계방법을 제안하였다. 적응진화알고리즘은 전역탐색특성이 우수한 유전알고리즘과 다음세대를 포함하는 해집단에 대해 적응적으로 우수한 국부탐색특성을 가진 진화전략을 사용한다. 재교배 과정에서 유전알고리즘과 진화전략을 위한 해집단의 분배는 적합도에 따라서 적응적으로 결정된다. 적응진화알고리즘은 퍼지제어기의 설계 파라메터인 퍼지변수에 대한 소속함수와 스케일 요소를 결정하는데 사용된다. 제기된 퍼지제어기의 성능을 평가하기 위해서 비선형 특성을 가진 실제 DC 모터 속도제어 시스템을 구성하여 실험하였으며, 실험결과 PD제어기의 경우보다 우수한 속도 제어성능을 가짐을 확인하였다.

ABSTRACT

In this paper, adaptive evolutionary algorithm(AEA) is proposed, which uses both genetic algorithm(GA) with good global search capability and evolution strategy(ES) with good local search capability in an adaptive manner, when population evolves to the next generation. In the reproduction procedure, proportion of the population for GA and ES is adaptively determined according to their fitness. The AEA is used to design membership functions and scaling factors of the fuzzy logic controller(FLC). To evaluate the performances of the proposed FLC design method, we make an experiment on the FLC for the speed control of an actual DC series motor system with nonlinear characteristics. Experimental results show that the proposed controller has better performance than PD controller.

키워드

Adaptive Evolutionary Algorithm (AEA), Fuzzy Logic Controller (FLC), DC series motors

Ⅰ. Introduction

During the last decade, fuzzy logic control has attracted a great attention from both the academic and industrial communities. Recently, fuzzy logic controller(FLC) has been suggested as an alternative approach to conventional control techniques for complex control system, such as a nonlinear or time delay system. That is, the design of FLC does not require a mathematical description of the control system and FLC can compensate the environmental

* 동명대학교 전기전자공학과

** 동서대학교 컴퓨터정보공학부

*** 동명대학교 항만물류학부

접수일자 : 2006. 10. 23

variation during o erating process [1][2]. However, we cannot obtain good control performances if the membership functions, fuzzy rules and scaling factors are incorrect. Recently, the membership functions, fuzzy rules and scaling factors are determined by evolutionary computations (EC), which is the probabilistic search method based on genetics and evolutionary theory [3][4].

There are three broadly similar avenues of investigation in EC: genetic algorithm (GA), evolution strategy (ES), and evolutionary programming (EP) [5]. GA simulates the crossover and mutation of natural systems, giving it a global search capability, whereas, ES simulates the evolution of an asexually reproducing organism.

The performance of EC is influenced by parameters such as size of population, fitness, probability of crossover and mutation etc.

If these are not adequately selected, the execution time will be longer and premature convergence to local minimum can occur. To solve the problems above, several approaches have been proposed. Arabas proposed an adaptive method of maintaining a variable population size[6]. The method introduces the concept of "age" of a chromosome, which is equivalent to the number of generations for which the chromosome stays "alive".

Srinivas proposed the adaptive genetic algorithm (AGA) in which the probabilities of crossover and mutation are varied depending on the fitness values of the solutions so as to maintain diversity in the population and sustain the ultimate convergence capacity of the GA[7].

To enhance the performances of ES and EP, mutation parameters are adapted during the run in ES and EP[8]. Schwefel proposed the method to self-adapt the mutation step size and the mutation rotation angles in ES[9]. Fogel introduced the idea of making the distribution of new trials from each parent an additional adaptive parameter[10].

In conventional method described above, parameter values and operator probabilities for the GA and ES are adapted to find solution and find it efficiently. In this paper, however, we proposed adaptive evolutionary algorithm (AEA), which is algorithm that ratio of population to which GA and ES will adapt is adaptively modified in the process of reproducing in according to fitness. We use ES to optimize locally, while the GA optimizes globally. In other words, the resulting hybrid scheme produces improved reliability by exploiting the "global" nature of the GA as well as the "local" improvement capabilities of the ES. AEA is used to design of the membership functions and scaling factors of FLC. The proposed FLC is applied to the speed control of an actual DC series motor system with nonlinear characteristics[11].

Ⅱ. Adaptive evolutionary algorithm

2.1. Motivation

In this paper, to reach the global optimum accurately and reliably in a short execution time, we designed an AEA by bringing together pieces of the GA and ES. In AEA, GA operators and ES operators are applied simultaneously to the individuals of the present generation to create the next generation. Individuals with higher fitness value have a higher probability of contributing one or more chromosomes to the next generation. This mechanism gives greater rewards to either the GA operation or the ES operation depending on what produces superior offspring.

2.2 Algorithm

In this paper, we adopted a (μ, λ) in ES. That is, only the λ offspring generated by mutation competes for survival and the μ parents are completely replaced in each generation. Also, self-adaptive mutation step sizes are used in ES.

For AEA to self-adapt use of GA and ES operators, each individual has an operator code to determine which operator to use. Suppose a '0' refers to GA, and a '1' to ES. At each generation, if it is more beneficial to use the GA, more '0's should appear at the end of individuals. If it is more beneficial to use the ES, more '1's should appear. After reproduction by roulette wheel selection according to the fitness, GA operations (crossover and mutation) are performed on the individuals that possess the operator code of '0' and the ES operation (mutation) is performed on the individuals that have an operator code of '1'. Elitism is also used. The best individual in the population reproduced both the GA population and ES population in the next generation. The major procedures of AEA are as follows:

1) *Initialization:* The initial population is generated randomly. For each individual, randomly initialize operator code. According to the operator code, GA operations are performed on the individuals with operator code '0', while ES operation is applied where the operator code is '1'.

2) *Evaluation and reproduction:* Using the selection operator, individual chromosomes are selected in proportional to their fitness, which is evaluated using an objective function. At every generation, the percentages of '1's and '0's in the operator code indicate the performance of the GA and ES operator.

3) *Preservation of minimum number of indivi- duals:* In this paper, we randomly change the operator code of the individuals with a higher percentage until the number of individuals for each EC operation becomes higher than a certain amount of individuals to be preserved. The predetermined minimum number of individuals to be preserved is set to 20% of the population size.

4) *Genetic algorithm and evolution strategy:* Modified simple crossover and uniform mutation are used as genetic operators. The modified simple crossover operator is the way to generate offstrings population, selecting two strings randomly in parents population, as shown in Fig. 1. If crossover occurs in k-th variable, selecting randomly two strings in t generation, offstrings of $t+1$ generation is shown in Fig. 1.

In uniform mutation, we select a random gene k in an individual. If an individual and the k-th component of the individual are the selected gene, the resulting individual is as shown in Fig. 2.

< Before Crossover > < After Crossover >

$S_v^t = [V_1, \ldots, V_k, \ldots, V_n]$ $\rightarrow$ $S_v^{t+1} = [V_1, \ldots, V_k', V_{k+1}' \ldots, V_n']$

$S_w^t = [W_1, \ldots, W_k, \ldots, W_n]$ $\rightarrow$ $S_w^{t+1} = [W_1, \ldots, W_k', W_{k+1}' \ldots, W_n']$

Crossover point

where, $V_j' = a_1 V_j + a_2 W_j$

$W_j' = a_1 W_j + a_2 V_j$

a_1, a_2 : Random numbers from [0, 1]

V_j : j-th gene of the vector S_v

W_j : j-th gene of the vector S_w

n : Number of parameters

Fig. 1. Modified simple crossover method

< Before Mutation > < After Mutation >

$S_v^t = [V_1, \ldots, V_k, \ldots, V_n]$ $\rightarrow$ $S_v^{t+1} = [V_1, \ldots, V_k', V_{k+1} \ldots, V_n]$

Mutation point

where, V_k' : Random value between upper bound and lower bound

Fig. 2. Uniform mutation method

Only the λ offspring generated by mutation operation competes for survival and the μ parents are completely replaced in each generation. Mutation is then performed independently on each vector element by adding a normally distributed Gaussian random variable with mean zero and standard deviation (σ), as Eq. (1). After adapting the mutation operator for ES population, if the improved ratio of individual number is lower than δ, the next generation standard deviation decreases in proportional to decrease rates of standard deviation (c_d), otherwise, the next generation standard deviation increases in proportional to increase rates of standard deviation (c_i), as shown in Eq. (2).

$$v_k^{t+1} = v_k^t + N(0, \sigma^t) \tag{1}$$

$$\sigma^{t+1} = \begin{bmatrix} c_d \times \sigma^t, & if & \phi(t) < \delta \\ c_i \times \sigma^t, & if & \phi(t) > \delta \\ \sigma^t, & if & \phi(t) = \delta \end{bmatrix} \tag{2}$$

$N(0, \sigma^t)$: Vector of independent Gaussian random variable with mean of zero and standard deviations

V_k^t : k-th variable at t generation

(t) : : Improved ratio of individual number after adapting mutation operator for population of ES in t generation

δ : Constants

5) *Elitism:* The best individual in a population is preserved to perform both GA operations and ES operation in the next generation. This mechanism not only forces GA not to deteriorate temporarily, but also forces ES to exploit information to guide subsequent local search in the most promising subspace.

Ⅲ. Design of fuzzy logic controller using adaptive evolutionary algorithm

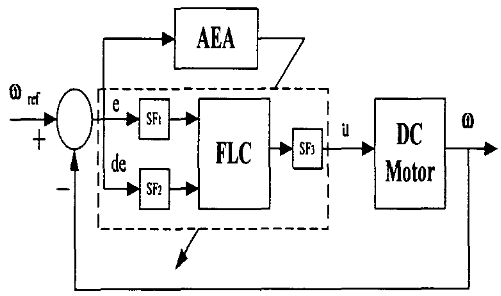
We tuned scaling factors of input/output and membership function shape of FLC using AEA. Fig. 3 shows architecture for tuning scaling factors of input/output and membership function shape of FLC using AEA. Speed deviation (e) and the change rate for speed deviation (de) are used as inputs of FLC as shown in Fig. 3. The FLC parameters used in this paper are given below.

- Number of input/output variables : 2/1
- Number of input/output membership functions:7/7
- Fuzzy inference method : max-min method
- Defuzzification method : center of gravity

Because deviation and change-of-deviation are used as inputs variables of the FLC, PD-like FLC is used. Rule base for the PD-like FLC from the two-dimensional phase plane of the system in terms of deviation and change-of-deviation is shown in Table 1.

When AEA's tuning the membership functions, fuzzy rules used for PD-type, as shown in Table 1, where linguistic variable NB means "Negative Big", NM means "Negative Medium", NS means "Negative Small" etc. Figurev 4 shows triangular membership function used in this paper.

In this paper, we fixed center of ZE to 0 and positive and negative membership functions are symmetrical to the 0. So the number of parameters of FLC will be 21, which means 3 centers and 4 widths for each variable as shown in Fig. 4.



e : speed deviation

de : change rate for speed deviation

u : control signal

SF_1, SF_2, SF_3 : scaling factors

Fig. 3. Block diagram of FLC using the AEA

Table. 1 PD-type fuzzy rules

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

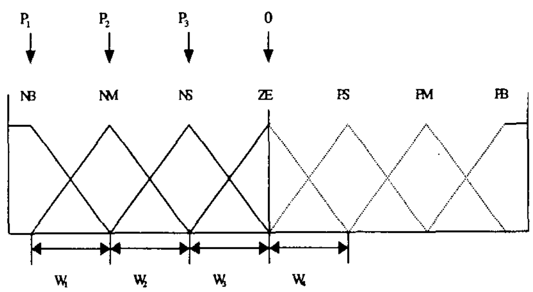
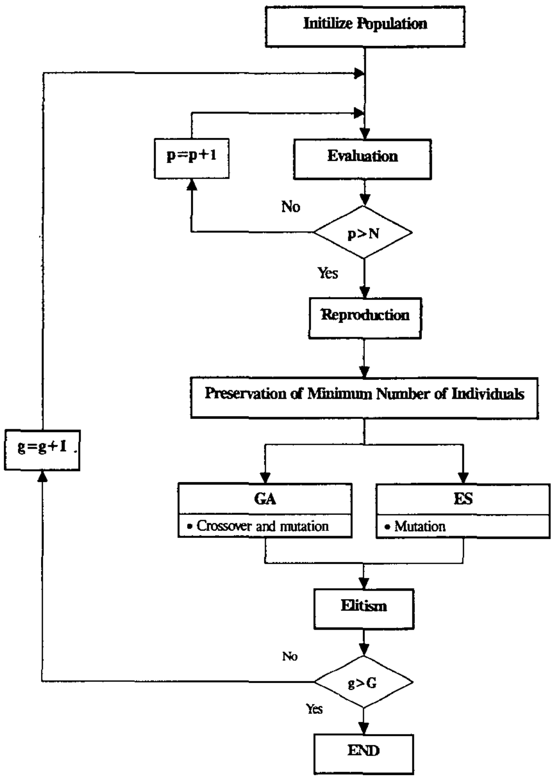


Fig. 4. Symmetrical membership functions

The flowchart for searching optimal solutions using the proposed AEA is shown in Fig. 5.



P : Number of population
G : Specified generation

Fig. 5. Flowchart for the design of FLC using AEA

The procedure of optimal solutions using AEA is as follows:

Step 1) Initialize population

The operation code is randomly set to decide whether each string is individuals of GA or individuals of ES. The configuration of population is shown in Fig. 6. Also scaling factors of the FLC are tuned using the AEA.

S_1	P_{11}	...	P_{19}	W_{11}	...	W_{112}	SF_{11}	SF_{12}	SF_{13}	*
S_2	P_{21}	...	P_{29}	W_{21}	...	W_{212}	SF_{21}	SF_{22}	SF_{23}	*
:										
S_n	P_{n1}		P_{n9}	W_{n1}		W_{n12}	SF_{n1}	SF_{n2}	SF_{n3}	*

n : population size
P_{ij} : center of the membership functions
W_{ij} : width of the membership functions
SF_{ij} : scaling factors
* : operator code

Fig. 6. Strings architecture for tuning membership functions and scaling factors

Step 2) Evaluation

We evaluated each string generated in Step 1) using the fitness function in Eq. (3). As shown in Eq. (3), the absolute error between an actual speed and the desired speed of an actual DC series motor is used.

$$Fitness = \frac{1}{100 + \sum_{k=1}^{N} |\omega_{c(k)} - \omega_{(k)}|} \tag{3}$$

where, ω : Actual speed
ω_c : Desired speed
N : No. of data acquired during T seconds

The procedure for evaluation part is as follows:

① We set initial values of deviation, change of deviation and control signal.
② We computed deviation and change of deviation between the desired speed and the actual speed.
③ We computed the output of FLC through fuzzy inference and defuzzification.
④ The control signal of FLC drives DC series motor through digital-to-analog converter.
⑤ The actual speed is measured, using tachometer of speed sensor.
⑥ We iterated ↑ and ∘ during experiment time, 18 [sec] (sampling time: 4 [msec]).

Step 3) Reproduction

We used roulette wheel reproduced in proportional to fitness. After reproducing, the individuals operator code of '0' is inserted in population of GA, the individuals operator code of '1' is inserted in population of ES.

Step 4) Preservation of minimum number of individuals

Among GA and ES, as one of which is stronger, we guaranteed minimum number of individuals to prevent offsprings from being disappeared by the remaining methods.

Step 5) GA and ES operation

The individuals operator code of '0' accomplishes crossover and mutation in GA operators and generates offsprings, and the individuals operator code of '1' accomplishes mutation in ES operator and generates offsprings.

Step 6) Elitism

We used elitism reproducing the best individual of fitness to GA and ES population by each one.

Step 7) Convergence criterion

We iterated Step 2) - Step 6) until being satisfied of the specified generation.

Ⅳ. Experimental results

Figure 7 shows the speed control system structure of the speed control of actual DC series motors. The AEA is used to optimize the shapes of the membership functions and scaling factors of FLC. The FLC and PD are implemented through C language in PC486, we used Lab Card to interface PC with speed signal and control signal. The sampling time of speed loop is 4 [ms] and that of current loop is 250 [μs].

The generator's parameter and motor's used in experiment are rated voltage (80 [V], 38 [V]), rated revolution (1000 [rpm], 800 [rpm]) and rated current (5.6 [A], 10 [A]), respectively. Figure 8 shows speed control system structure of the speed control of an actual DC series motor. Table 2 shows the simulation parameters of AEA for tuning FLC. Figure 9 shows the shapes of the membership functions tuned by AEA. Error rate and membership function of output are symmetric to membership function of ZE.

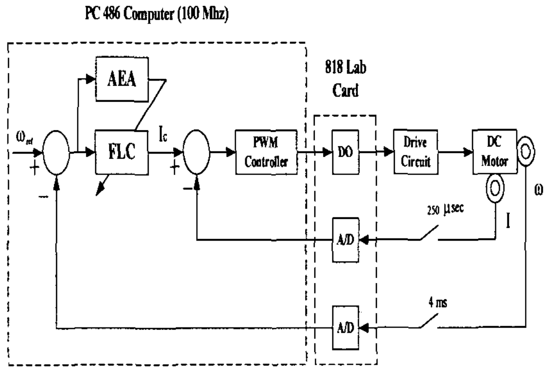


Fig. 7. Laboratory setup for DC series motor speed control

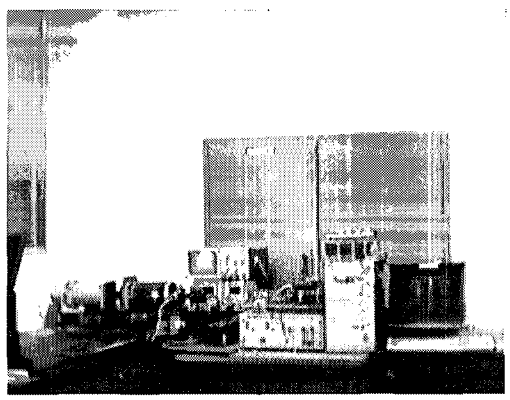
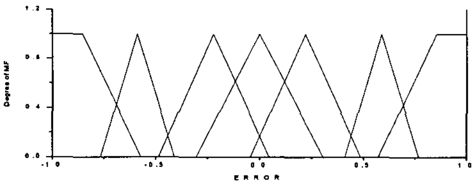
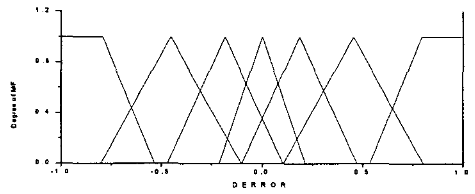


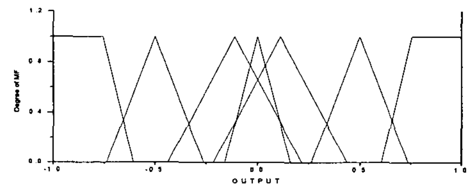
Fig. 8. Experimental apparatus of a DC series motor system



(a) The membership function of error



(b) The membership function of error rate



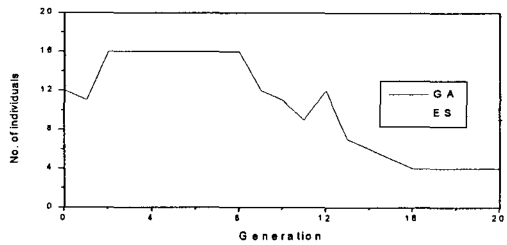
(c) The membership function of output

Fig. 9. Tuned membership functions using AEA

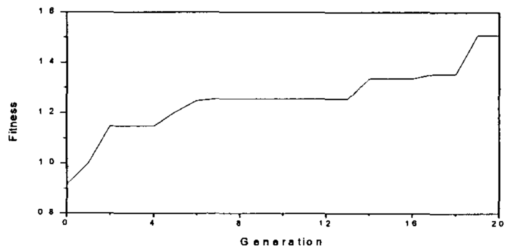
Table. 2 Simulation parameters used in the AEA

Methods	AEA
Size of population	20
Crossover probability	0.85
Mutation probability	0.05
δ	0.5
C_d	0.85
C_1	1.15

Figure 10 (a) shows the fitness values by the AEA in each generation. Fig. 10 (b) shows the number of individuals for GA and ES in the AEA. The number of individuals of GA is higher than that of individuals of ES in early generation. But, from generation to generation, the number of individuals of ES is higher than that of individuals of GA. The AEA produces improved reliability by exploiting the "global" nature of the GA initially as well as the "local" improvement capabilities of the ES from generation to generation.



(a) Fitness value



(b) Number of individuals of GA and ES in AEA

Fig. 10. Fitness functions and number of individuals of GA and ES

The desired speed used in tuning FLC, using AEA, needed nonlinear function such as in Eq. (4), and Eq. (5) and Eq. (6) are the desired speed used in evaluating the robustness of tuned FLC by AEA, where T is sampling time.

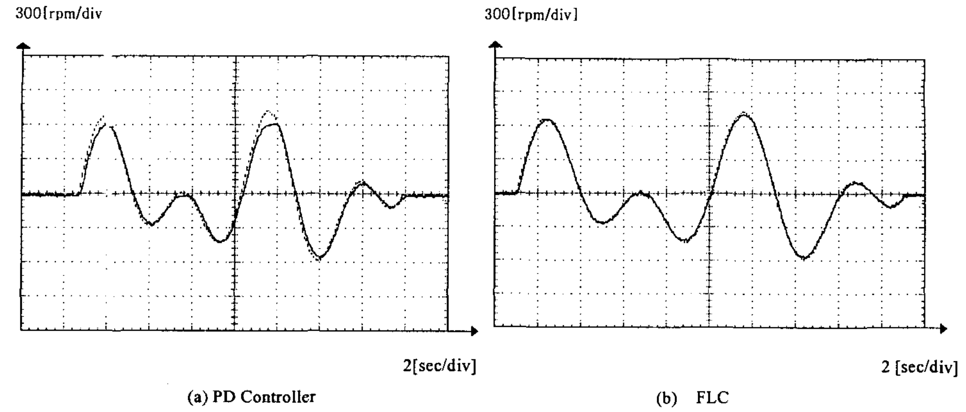
$$\omega_c(k) = 300\sin(2\pi kT / 3) + 300\sin(2\pi kT / 5) \quad (4)$$

$$\omega_c(k) = 300\sin(2\pi kT / 3) + 300\sin(2\pi kT / 7) \quad (5)$$

$$\omega_c(k) = 300\sin(2\pi kT / 4) + 300\sin(2\pi kT / 7) \quad (6)$$

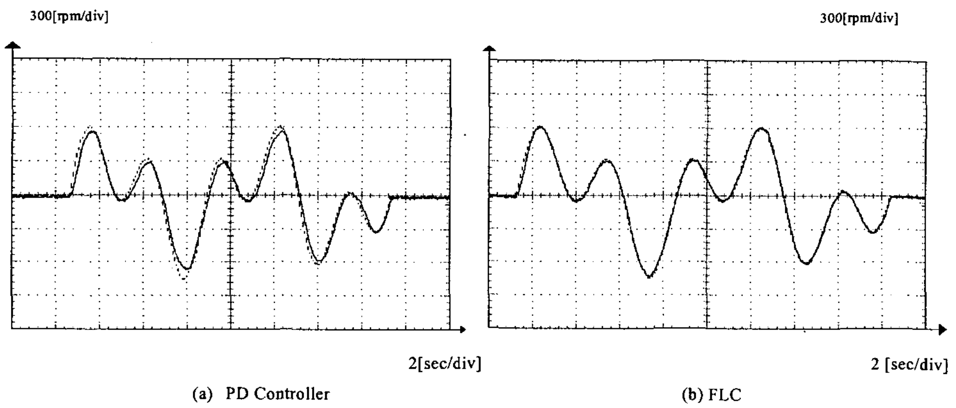
Figure 11 shows the experimental result of DC series motor system for the desired speed in Eq. (4) used in tuning AEA.

And the proportional-derivative gain of PD controller was determined using AEA. When the characteristic is optimal about the reference signal at real time, the value of gain is used as the gain of PD controller. Therefore the



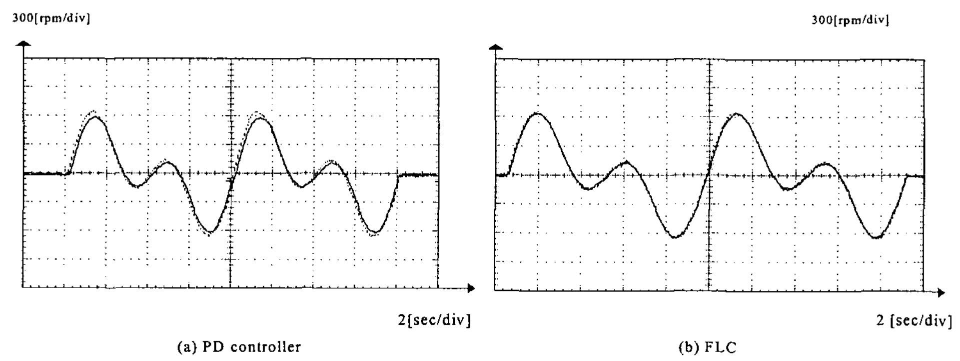
-- : Desired speed, — : Actual speed　　-- : Desired speed, —: Actual speed

Fig. 11. Comparisons of speed response with PD controller and FLC



-- : Desired speed, — : Actual speed　　-- : Desired speed, —: Actual speed

Fig. 12. Speed response with new reference speed



-- : Desired speed, — : Actual speed　　-- : Desired speed, —: Actual speed

Fig. 13. Speed response with new reference speed

proportional- derivative gain of PD controller was optimized by the AEA, and dot line represents the desired speed and straight line actual speed.

The proposed FLC produces more accurate speed response than PD controller in terms of tracking performance. Therefore, the proposed FLC demonstrates the better tracking performance, compared with the PD controller.

To evaluate the robustness of the FLC, FLC is also tested over new desired speed in Eq. (5) and Eq. (6), which is not used when tuning. Figure 12 and Figure 13 show the experimental result for the desired speed in Eq. (5) and Eq. (6). The tracking performance gets to be deteriorated, for a new desired speed of PD controller. But the proposed FLC of the paper shows a better tracking performance in both peak value and the others for the new desired speed. From the experimental results confirm that FLC shows better performance than PD controller over another new desired speed.

V. Conclusion

In this paper, we proposed the adaptive evolutionary algorithm as the new computation method having global and local search capability in an adaptive manner when population evolves to the next generation. The proposed AEA is used in designing of the membership functions and scaling factors of FLC. And the designed FLC is applied to the speed control of an actual DC series motor system with nonlinear characteristics. Experimental results show that FLC has better control performance than PD controller in terms of rising time, settling time.

Also FLC has better performance over another new desired speed. In the early generation, it is shown the number of population of GA is higher than that of population of ES and the number of population of ES gets more higher as number of generation increases.

This shows global search is executed by GA in the early generation and local search is executed adaptively by means of ES as the number of generation increases.

References

[1] Abraham Kandel, Gideon Langholz, "Fuzzy Control Systems", CRC Press, 1994

[2] Y. S. Kung and C. M. Liaw, "A Fuzzy Controller Improving a Linear Model Following Controller for Motor Drives," *IEEE Transactions on Fuzzy Systems*, vol. 2, no. 3, pp. 194-201, Aug., 1994.

[3] Jinwoo Kim, Yoonkeon Moon, and Bernard P. Zeigler, "Designing Fuzzy Net Controllers Using Genetic Algorithms", *IEEE Transaction on Control System*, pp. 66-72, June, 1995

[4] K. Krishnakumar, David E. Goldberg, "Control System Using Genetic Algorithms", *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 3, pp. 735-740, May, 1992

[5] D. A. Linkens and H. O. Nyongesa, "Genetic Algorithms for Fuzzy Control: Part 1 : Offline System Development and Application", *IEE Proceeding Control Theory Application*", Vol. 142, No. 3, pp. 161-176, May, 1995

[6] J. Arabas, Z. Michalewicz, and J. Mulawka, "GAVaPS-a Genetic Algorithm with Varying Population Size", *IEEE International Conference on Evolutionary Computation*, pp. 73-78, 1994

[7] M. Srinivas and L. M. Patnaik, "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 24, No. 4, pp.656-667, April, 1994

[8] D. B. Fogel, L. J. Fogel and J. W. Atmas, "Meta-Evolutionary Programming", *Proceedings 2sth Asilomar Conference on Systems, Signals, and Computers*, pp. 540-545, 1991

[9] H. P. Schwefel, "Evolution and Optimum Seeking", *Sixth Generation Computer Technology Series*, Wiley, 1995

[10] D. B. Fogel, "An Introduction to Simulated Evolutionary Optimization", *IEEE Transactions on Neural Networks*, Vol. 5, No. 1, Jan. 1994

[11] G. H. kim, S. I. Kang, and J. H. Lee, "Study on the Real-Time Moving Object Tracking using Fuzzy Controller", Journal of KIMICS, Vol 10, No. 1, pp. 191-196, 2006

저자소개

Dong-Wan Kim

received his M. S. and Ph. D. degrees in electrical engineering from Pusan National University. He is currently a professor at Tongmyong University.
His research interest is in applications of robust control, TDF control and intelligent control.

Gi-Hyun Hwang

received his M.S. and Ph.D. degrees in electrical engineering from Pusan National University. He is currently a professor at Dongseo University.
His research interest is in applications of intelligent control to power system.

Jae-Hyun Lee

received his M.S. and Ph.D. degrees in electronics and communication engineering from Korea Maritime University. He is currently a professor at Tongmyong University. His research interest is in applications of Human-Interface and artificial intelligence control.