타부 탐색법과 신경 회로망을 이용한 적응 퍼지제어기의 설계

論 文
50P-3-6

Design of Adaptive Fuzzy Logic Controller Using Tabu Search and Neural Network

黃 琪 鉉* · 金 烔 完** · 李 東 熙*** · 金 亨 洙§ · 申 東 律§§ · 申 春 植§§§
(G. H. Hwang · D. W. Kim · D. H. Lee · H. S. Kim* · D. L. Shin · C. S. Shin)

1. Introduction

Fuzzy logic controller (FLC) has attracted a great attention from both the academic and industrial communities. Recently, 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 variation during operating process [1,2]. However, we cannot obtain a good control performance if the membership functions, fuzzy rules and scaling factors are incorrect. Recently, the membership functions, fuzzy rules and scaling factors are determined by evolutionary computation (EC), which is the probabilistic search method based on genetics and evolutionary theory [3-8].

Tabu Search, which had been first studied in the middle of 1980s by Glover, less converges to local minimum than general numerical method, and also is easier to use the knowledge of the corresponding problem than the global optimization searches such as simulated annealing and genetic algorithm. The conventional optimization algorithm has a higher complexity of the corresponding problem and costs too much computation time in converging to the best solution in case that the search region is wide. But Tabu Search has a high convergence speed with involving directly itself in generation of candidate solution and searching the complex problem efficiently. And it has been studying in many directions actively because of its easy combination with other optimization algorithm [9-13].

The optimal search capability of Tabu Search is affected by initial solution, neighbor selection, and Tabu list size. Generally, initial solution is selected based on the expert experience or the numerical method because its erroneous selection can make itself converge to the local minimum. The definition and size of neighbor are dependent on the object or search strategy. The erroneous neighbor selection can make the search region larger and computation time longer owing to an unnecessary searching region. In this paper, we proposed an improved Real-type Tabu search (RTS) changing adaptively the neighbors range to be searched according to the object function each iteration after generating the neighbors for the current solution according to search strategy or object within searching region and evaluating them according to the object function.

We designed an adaptive fuzzy logic controller (AFLC) for speed control of DC servomotor using the proposed RTS and forward neural network. In this, we generated the neighbors for the current solution according to search strategy or object within search region and evaluated them according to the object function. We proposed RTS changing adaptively the range of neighbor to be searched according to the object function each iterations. The design of AFLC for speed control of DC servomotor using the improved RTS and forward neural network is composed of two steps. The first step is for optimizing the weights of the forward neural network and the initial value of scaling factors needed in real-time adjusting scaling factors of FLC by the forward neural network using the improved RTS. The second step is for real-time changing output scaling factor of FLC using the forward neural network. The proposed AFLC was applied to the speed control of an actual DC servomotor system.

* 正 會 員 : 釜山大 컴퓨터 및 情報通信硏究所 基金敎授
** 正 會 員 : 東明大學 電氣·電子 系列 助敎授
*** 正 會 員 : 釜山大 電氣工學科 博士
§ 正 會 員 : 釜山大 電氣工學科 博士科程
§§ 正 會 員 : 청송산업(주) 先任硏究員·博士
§§§ 正 會 員 : 부경대 制御計測工學科 講師·博士
接受日字 : 2001年 6月 12日
最終完了 : 2001年 9月 7日

Experimental results show that AFLC showed the better control performance than PI controller in terms of settling time, rising time and overshoot.

2. Design of AFLC Using RTS and Neural Network

2.1 The Advantage of RTS

In this paper, we proposed an RTS improving the optimal solution search capability. The proposed RTS adaptively changes the range of the neighborhood solution according to the objective function each iteration. The procedure of the proposed RTS is composed of generating initial solution, selecting and evaluating neighborhood solution, and generating trial solution, Tabu Test, and aspiration condition. The detail of them is explained in next section.

The RTS, as an empirical search method, rarely falls into the local minimum, compared with general variational one. Also it effectively uses the information obtained during searching and the knowledge of corresponding object, compared with the global search methods such as genetic algorithm and simulated annealing. The conventional optimization algorithm is a very costly process in converging to the optimal solution in case of high complexity and wide search space. Besides, the RTS is capable of searching effectively about the complex problem such that rapid converging. It combines with other optimization algorithms easily.

2.2 Tuning Scaling Factors of FLC and Weights of Neural Network Using RTS

Generally, FLC shows a good control performance against disturbance and parameter change of control system, but does not against various disturbances and parameter changes of control system. So we designed an intelligent adaptive FLC to change adaptively scaling factors of FLC using the forward neural network each sampling time, which is to obtain the optimal control performance against the various disturbances and parameter change of control system. We tuned the weights of the forward neural network and scaling factors of FLC using RTS, the empirical searching method, by means of repeating learning. Fig. 1 shows the structure for tuning the weighs of the forward neural network and scaling factors of FLC using RTS. The inputs of FLC are error multiplied by scaling factor and error deviation multiplied by scaling factor in (1), (2). The inputs of the forward neural network are errors having the time delay. In this paper, we used an isosceles triangle method as fuzzifier, Mamdanis max-min method as a fuzzy reasoning, the center of gravity as defuzzifier, a proportional-differential type as fuzzy rule. The structure of neural network is composed of input, hidden, and output layer. The number of neuron of input, hidden, and output layer is 3, 5, 1 respectively.

$$E(t) = SF_1^{RTS}\, e(t) \tag{1}$$

$$DE(t) = SF_1^{RTS}\, \frac{de(t)}{dt} \tag{2}$$

where, $e(t) = Ref(t) - Out(t)$

SF_1^{RTS}: Error scaling factor of FLC tuned RTS

SF_2^{RTS}: Error rate scaling factor of FLC tuned RTS

$Ref(t)$: Reference speed of motor

$Out(t)$: Actual speed of motor

The absolute sum of error is used as the input of RTS used for learning the weights of the forward neural network and scaling factors of FLC. The deviation of output scaling factor of FLC in (3) is used as the output of the forward neural network. It is found the output scaling factors of FLC are changed adaptively by forward neural network each sampling time in (3). We can obtain the optimal control performance on the changing control system by changing the scaling factors of FLC each sampling times using such a forward neural network.

$$SF_3(t) = SF_3^{RTS} + \triangle SF_3(t) \tag{3}$$

where, $SF_3(t)$: Scaling factor of FLC

SF_3^{RTS} : Scaling factor of FLC tuned by RTS

$\triangle SF_3(t)$: Scaling factor of FLC changing real-time by forward neural network

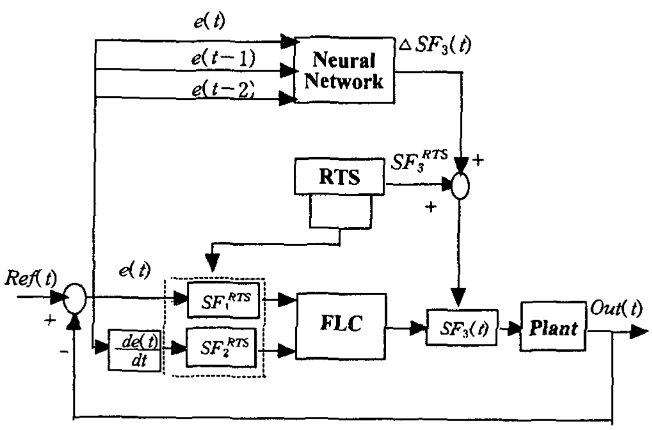


Fig. 1 Configuration for tuning scaling factors of FLC and weights of neural network using RTS

The design procedures of scaling factors of FLC and weights of the forward neural network using the proposed RTS are composed of initial solution selection, neighborhood decision and evaluation, trial solution generation, Tabu list, and aspiration criterion. The design procedure of scaling factors of the FLC and weights of neural network is as follow. The flow chart is shown in **Fig. 2**.

Step 1) Selection of initial solution: Generating the initial solution like (4) such that the constraint condition of the given problem is satisfied, and then setting it as the current solution.

$$S_0 = random[\{SF_i^{max},\ SF_i^{min}\},\ \{\omega_i^{max},\ \omega_i^{min}\}] \quad (4)$$

where, SF_i^{max} : i-th upper bound of scaling factors of FLC

SF_i^{min} : i-th lower bound of scaling factors of FLC

ω_i^{max} : i_th Upper bound of weights of neural network

ω_i^{min} : i_th lower bound of weights of neural network

Step 2) Neighborhood decision and evaluation: Generating the neighbor solution for the current solution according to search strategy and object within the search region, and then evaluating the object function set like (5).

$$J = Minimize \sum_{k=1}^{N} E(k) \quad (5)$$

where, N : No. of data acquired during specified time

Fig. 3 shows method for generating neighbor. The range of neighbor solution to be searched for the initial solution satisfying the restraint condition is adaptively modified each iteration. The procedure modified adaptively each iteration in Fig 3 is given as (6). As shown in (6), objective values with evaluating the neighbor solution generated in t-th iteration are found and then the standard deviation for an upper 10[%] of the objective function are found. And the neighbor solution range (Rt+1) in t+1-th iteration makes the neighbor solution range (Rt) increase at an increasing rate of Ci if the standard deviation is smaller than the specified standard deviation, but the neighbor solution range (Rt) decrease at a decreasing rate of Cd, otherwise.

$$R^{t+1} = \begin{bmatrix} C_d \times R^t & \text{if } \delta(t) < K \\ C_i \times R^t & \text{if } \delta(t) > K \\ R^t & \text{if } \delta(t) = K \end{bmatrix} \quad (6)$$

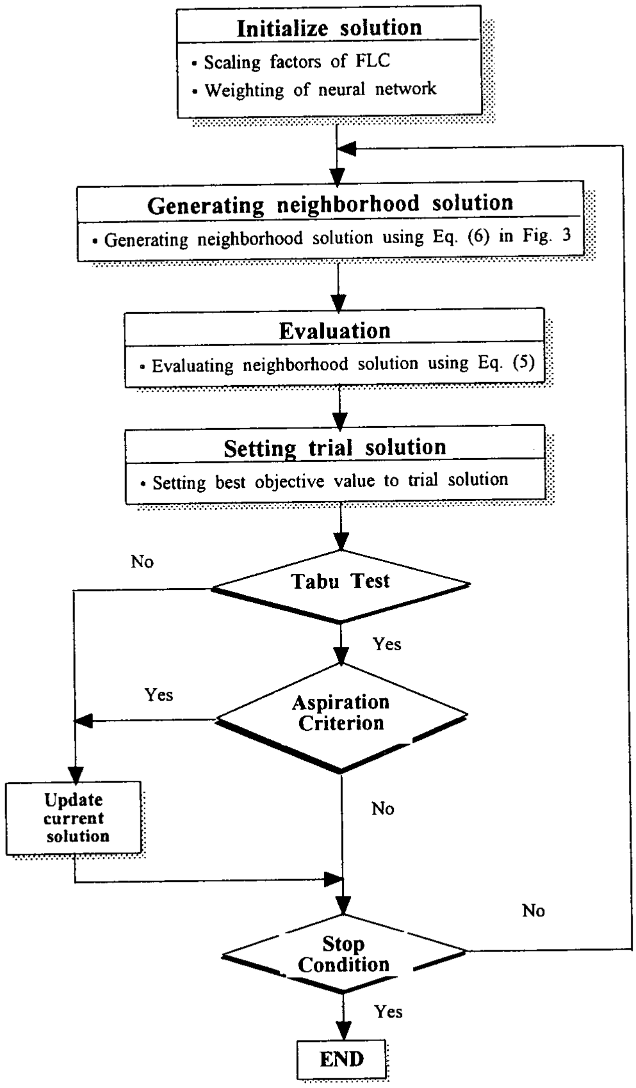


Fig. 2 Flowchart of scaling factors of FLC and weights of the forward neural network using the proposed RTS

where, R^t : Range of neighbor solutions at t-th iteration

R^{t+1} : Range of neighbor solutions at t+1-th iteration

$\delta(t)$: The standard deviation for above 10[%] out of objective function of neighbor solutions in t-th iteration

C_d, C_i : Decreasing and increasing rates of range of neighbor solutions

K : The specified standard deviation

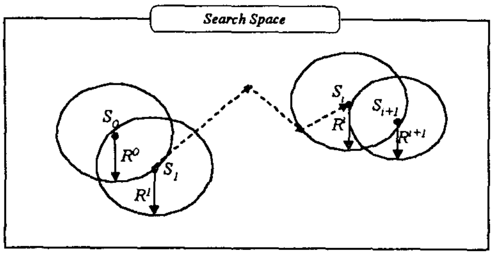


Fig. 3 Neighborhood decision

Step 3) Trial Solution: Setting the best solution of the evaluated values in the neighbor solution generated in Step 2) as a trial solution.

Step 4) Tabu Test: Examining whether the trial solution is included with Tabu list or not. In this paper, the Tabu list is composed of the best solution searched each iteration, we used first in first out (FIFO) method. If a Euclidean distance from the solution with best objective values of the neighbor solution to the best solution searched up to now is below the definite value, we set the solution as Tabu. If the trial solution is included within Tabu list, perform Step 5) or Step 6).

Step 5) Aspiration criterion: If the aspiration criterion is satisfied through the trial solution is included with Tabu list, the trial solution is excluded and set as the current solution in the next search. In this paper, when the trial solution is superior to the best solution searched up to now through it is Tabu, we used it as the aspiration criterion.

Step 6) Current solution: Setting trial solution as the current solution of next search if it is not included within Tabu list.

Step 7) Stop condition: Repeating Step 2) Step 6) until specified iteration.

2.3 Design of AFLC using the forward Neural Network

Generally FLC shows a good control performance against parameter deviation of the controlled and disturbance, but not an optimal control performance. In this view, to obtain an optimal control performance against parameter deviation of the controlled and various disturbances, we proposed a method adaptively changing the output gain of fuzzy controller each sampling time in real time using a neural network. The configuration is shown in Fig. 4. The output of neural network is the deviation of output gain of FLC. The weights of the forward neural network are ones tuned by RTS.

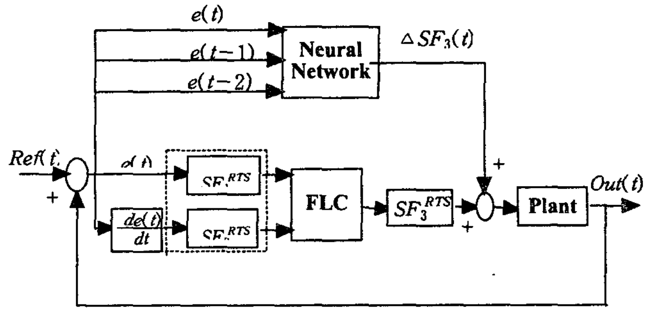


Fig. 4. Configuration of AFLC using the forward neural network

3. Experimental Results

Fig. 5 shows the speed control system structure for the speed control of an actual DC servomotor. As shown in Fig. 5, the improved RTS is used to optimize the scaling factors of FLC and weights of forward neural network. FLC is implemented through C language in PC586, we use Lab Card to interface PC with speed signal and control signal. The sampling time of speed loop is 4ms and that of current loop is 250s. The parameters of generator and motor used in our experiment are rated voltage (75 [V], 50 [V]), rated revolution (1500 [rpm], 800 [rpm]) and rated current (3.5 [A], 3 [A]), respectively. Fig. 6 shows speed control system structure for the speed control of an actual DC series motor. Table 1 shows the simulation parameters of RTS for tuning scaling factors of FLC and the forward neural network. It takes one hour in tuning FLC optimally in the simulation of RTS, as shown in the Table 1. Fig. 7 shows the objective values by the RTS each iteration. As shown in Fig. 7, the object function decreases as the number of iteration increases. This means FLC and NN are optimized.

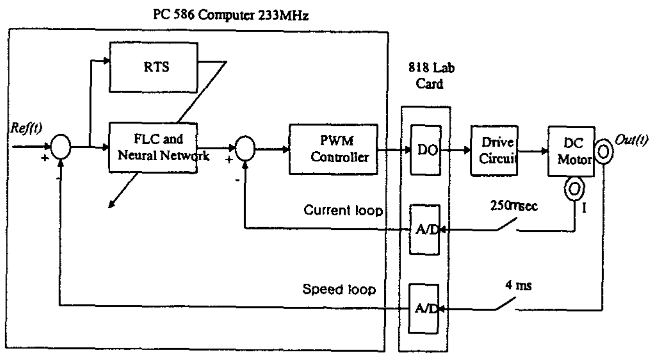


Fig. 5 Laboratory setup for DC servo motor speed control

Table I. Coefficients for simulation using RTS.

Methods		RTS
Size of neighbor solution		10
No. of tabu list		5
Tabu criterion		0.05
Number of iteration		30
Neighbor solution range	Decreasing rate (C_d)	0.95
	Increasing rate (C_i)	1.05
	K	0.001

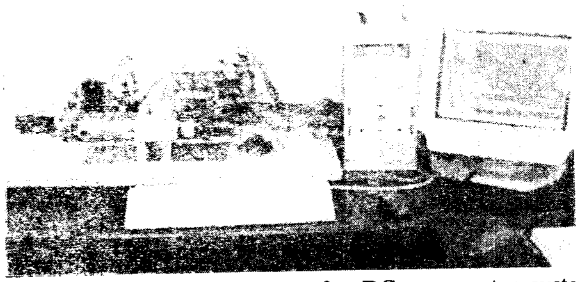


Fig. 6 Experimental apparatus of a DC servomotor system

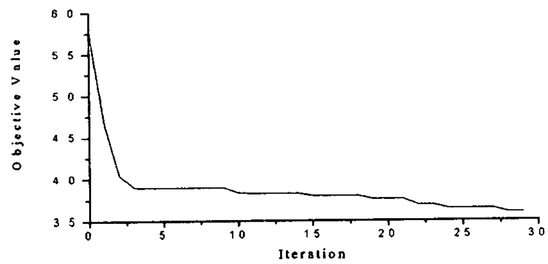


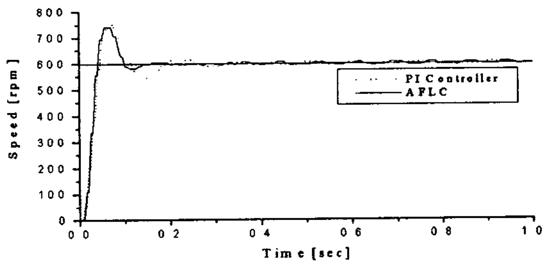
Fig. 7 The changing objective value each iteration

Fig. 8(a) shows the experimental results of the DC servomotor for reference speed used when RTS tunes AFLC and PI controller. As shown in Fig. 8(a), the proposed AFLC produced the better control performance than PI controller in terms of rising time and settling time. Fig. 8(b) shows the changing scaling factor of AFLC each sampling time. To evaluate the robustness of the AFLC, we also tested AFLC over a new reference speed that had not used when tuning. As shown in Fig. 9-Fig. 10, the experimental results confirmed that AFLC shows the better performance than PI controller over rising time, settling time, and overshoot. Fig. 11 shows the response characteristic curve when a resistance load (about 30[Ω]) applies at 1[sec]. As shown in Fig. 11, AFLC showed a good control performance against changing load.

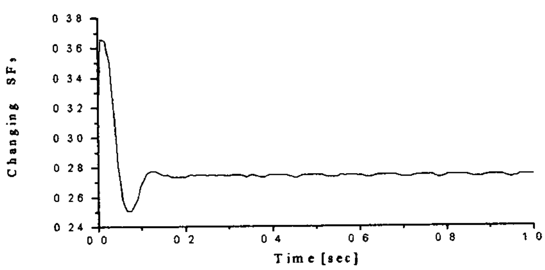
Table II. Comparisons of control performances of PI and AFLC

Methods \ Speed		Reference Speed 400[rpm]	Reference Speed 600[rpm]	Reference Speed 800[rpm]
Rising time [sec]	PI	0.03	0.03	0.03
	AFLC	0.02	0.02	0.02
Settling time [sec]	PI	0.3	0.25	0.25
	AFLC	0.15	0.12	0.15
Overshoot [%]	PI	25	25	23
	AFLC	12	21	10

The results are summarized in Table 2. AFLC shortened the rising time by 0.01[sec] and reached the steady state faster by 0.1[sec] ~ 0.15[sec] in terms of settling time and lowers overshoot by 5[%] ~ 13[%]. As showen in Table Ⅱ, AFLC showed a good control performance in terms of settling time. When reference speed was 600[rpm], AFLC was similar to the performance of PI controller in terms of overshoot. When reference speeds were 400[rpm] and 800[rpm], AFLC showed the better performance then PI controller.

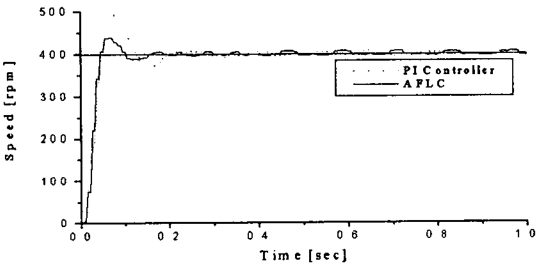


(a) Response Speed

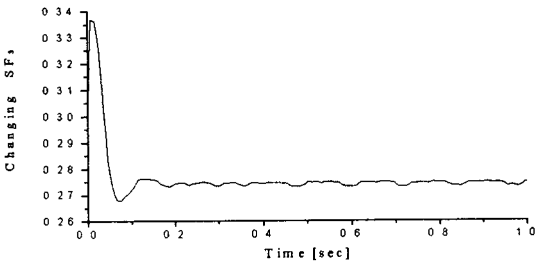


(b) Changing SF3

Fig. 8 Responses of DC servomotor when the speed was 600 [rpm]

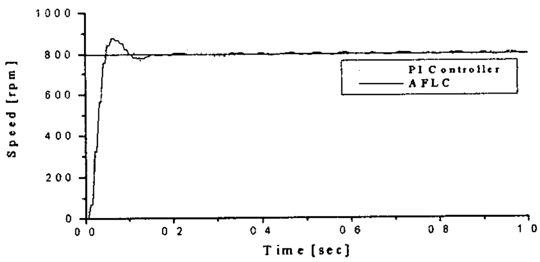


(a) Responses Speed

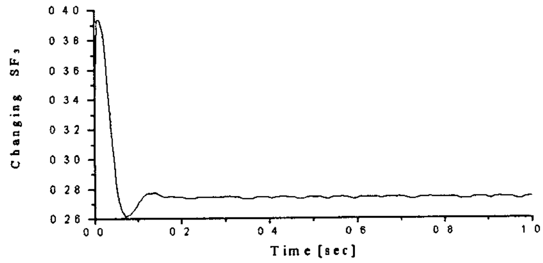


(b) Changing SF3

Fig. 9 Responses of DC servomotor when the speed was 400 [rpm]



(a) Responses Speed



(b) Changing SF3

Fig. 10 Responses of DC servomotor when the speed was 800 [rpm]

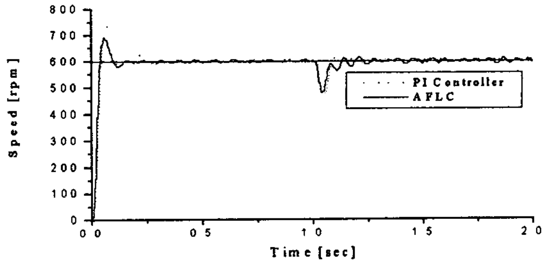


Fig. 11 Responses of DC servomotor when the load was changed

4. Conclusions

We proposed the AFLC for speed control of DC servomotor using the proposed RTS and forward neural network. The AFLC adaptively modified the output scaling factor of FLC each sampling time. To evaluate the usefulness of the proposed method, it was applied to the speed control of an actual DC servomotor system. The experimental results showed that AFLC has the better control performance than PI controller in terms of settling time, rising time and overshoot toward the reference speed used in tuning AFLC. To evaluate the robustness of AFLC, we test over a new reference speed and changing load not used when tuning. AFLC shows the better control performance than PI controller.

References

[1] A. Kandel, G. 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] K. Krishnakumar, D. E. Goldberg, Control System Using Genetic Algorithms, Journal of Guidance, Control, and Dynamics, Vol. 15, No. 3, pp. 735-740, May, 1992

[4] J. J. Buckley and Y. Hayashi, Fuzzy Genetic Algorithms for Optimization, Proceeding of 1993 International Joint Conference on Neural Networks, pp. 725-728, 1993

[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] D. A. Linkens and H. O. Nyongesa, Genetic Algorithms for Fuzzy Control: Part 2 : Online System Development and Application, IEE Proceeding Control

Theory Application, Vol. 142, No. 3, pp. 177-185, May, 1995
[7] J. Kim, Y. Moon, and Bernard P. Zeigler, Designing Fuzzy Net Controllers Using Genetic Algorithms, IEEE Transaction on Control System, pp. 66-72, June, 1995
[8] A. Homaifar and E. McCormic, Simulataneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Usng Genetic Algorithms, IEEE Transactions on Fuzzy Systems, Vol. 3, No. 2, pp. 129-139, May, 1995
[9] F. Glover, "Tabu Search-Part I," ORSA Journal on Computing Vol. 1. No. 3, pp. 190-206, 1989.
[10] F. Glover and M Laguna, "Tabu Search", Kluwer Academic Publishers, pp. 111-116, 1997
[11] M. D. Asic and V. V. Kovacevic-Vujcic, "TABU Search Methodology in Global Optimization", Computers and Mathematics with applications 37, pp. 125-133, 1999
[12] Y. Shiyou, N. Guangzheng, L. Yan, T. Baoxia and L. Ronglin, "An Universal Tabu Search Algorithm for Global Optimization of Mulitmodal Functions with Continuous Variables in Electromagnetics", IEEE Trans. on Magnetics, Vol. 34, No. 5, pp. Sep. 1998
[13] H. Chun, H. Kim, G. Hwang, K. Mun and J. H. Park, "Thermal Unit Commitment Using Tabu Search", Transaction on KIEE, Vol. 49A, No. 2, pp. 70-77, Feb., 2000
[14] J. Nam, H. Ahn, G. Hwang, D. Kim and D. Lee, "Design of Adaptive Fuzzy Logic Controller Using Real-Coding Genetic Algorithm and Neural Network", Transaction on KIEE, Vol. 50P, No. 1, pp. 35-42, Mar., 2001

감사의 글

본 논문은 중소기업청에서 지원한 2001년도 중소기업청 컨소시엄연구과제 결과임. 과제번호(TMC-7)

저 자 소 개

황 기 현(黃 琪 鉉)

1968년 3월 1일생. 1994년 경성대 공대 전기공학과 졸업. 1996년 부산대 일반대학원 전기공학과 졸업(석사). 2000년 동 대학원 전기공학과 졸업(박사). 현재 부산대 컴퓨터 및 정보통신 연구소 기금교수. 현재 연구분야는 지능제어, 진화연산, Tabu Search, 전력제어 및 FACTS 등.

김 동 완(金 烔 完)

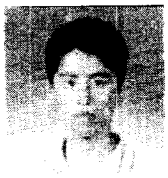
1960년 2월 1일생. 1984년 동아대 전기공학과 졸업. 1987년 부산대 대학원 전기공학과 졸업(석사). 1995년 동 대학원 전기공학과 졸업(박사). 1997년 ~ 현재 부산대학교 생산기술연구소 연구원. 동명대학 전기·전자계열 조교수. 관심분야-2자유도제어, 강인제어, 지능제어.

Tel : 051-620-3416

E-mail : dongwan@yongma.tmc.ac.kr

이 동 희(李 東 熙)

1970년 11월 11일생. 1996년 부산대 공대 전기공학과 졸업. 1998년 동 대학원 전기공학과 졸업(석사). 2001년 동 대학원 전기공학과 졸업(박사) 산업전자 및 전력전자 전기기기 분야에 관심



김 형 수(金 亨 洙)

1972년 1월 26일 생. 1994년 부산대공대 전기공학과 졸업. 1997년 부산대 일반대학원 전기공학과 졸업(석사). 현재 동 대학원 박사과정

Tel : (051) 510 - 3188

E-mail : kimhsu@hyowon.pusan.ac.kr

신 동 률(申 東 律)

1970년 12월 25일생. 1995년 동아대 전기공학과 졸업. 1997년 동아대 대학원 전기공학과 졸업(석사). 1997년-현재 동아대 대학원 전기공학과 박사과정 수료. 관심분야는 전력전자, 벡터제어, 제어응용, 지능제어 등.

Tel : 051) 200-7734

E-mail : g9775996@donga.ac.kr

신 춘 식(申 春 植)

1960년 8월생. 1987년 부경대학교 전자공학과 졸업(공학사). 1992년 동아대학교 대학원 전자공학과 졸업(공학석사). 1997년 한국해양대학교 대학원 기관공학과 졸업(공학박사).

2000년~현재 부경대학교 제어계측공학과 강사. 관심분야 : 제어공학

Tel : 011-865-2635

E-mail : shincs85@dreamwiz.com