IEEE International Workshop on Intelligent Robots and Systems
IROS '90

APPLICATION OF A SELF-ORGANIZING FUZZY CONTROL TO THE JOINT CONTROL OF A PUMA-760 ROBOT

K.C. Koh*, H.S. Cho*, S.K. Kim** and I.S. Jeong**

* Department of Production Engineering, Korea Advanced Institute of Science & Technology
P.O. Box 150, Chongryangri, Seoul, Korea

** Robotics R & D Team, Samsung Electronics Co. Ltd. Suwon, Korea

ABSTRACT: Utilization of the fuzzy self organizing control(SOC) algorithm for the control of robotic manipulators is becoming increasingly of interest. However, the control performance of the present self-organizing control(SOC) algorithm is often limited due to quantization of process variables and requires still too large amount of computation time to be executed within small sampling time. To improve these limitations this paper presents a combined control structure in which the conventional P-control law is used in parallel with a modified fuzzy SOC algorithm. To run the algorithm fast, the inference mechanism of the fuzzy rules is simplified by modifying the SOC algorithm in [5]. Addition of the P-control law is made in the feedforward path to guarantee stability at initial stage before the rules are completely created. To test the performance of the proposed control scheme and structure, a series of experiments were performed using a PUMA-760 robot. The experimental results show that the proposed structure can be effectively applied to the control of robot manipulators when implemented to the personal computer.

1. INTRODUCTION

Robot manipulators are widely used for the process automation in the industry and their application areas are being extended to complex tasks requiring more strict control performances with some intelligences. In the design of the joint controllers for such robots, there have been difficulties due to model uncertainties and changes in control circumstance. Therefore, the control algorithms are required to be more sophisticated in order to cope with more complex control environments. The adaptive control schemes can be the approaches to that, but their algorithms are usually too complex to be computed in real time. It seems that a more intelligent and fast control scheme needs to be developed.

In recent years, for the AI approach to automatic control, the fuzzy rule-based control schemes have been applied to the control of the complex processes or ill-defined uncertain processes. The control of robotic manipulators is one of such areas which enjoy the essence of the control algorithm. This is due to the fact that robotic manipulators posses the complex, uncertain and nonlinear dynamic characteristics which make it difficult to derive an exact mathematical model. Furthermore, the fuzzy control scheme does not require so large amounts of computation time as in the traditional adaptive algorithms and can easily realize the heuristic rules obtained from human experiences which can not be expressed in the mathematical form. Practical applications of the fuzzy controllers can be found from various fields such as a flexible arm control[1], servo motor control[2], trajectory planning for obstacle avoidance[3], and automated guided vehicle[4].

The simple fuzzy control schemes applied to these problems have a demerit in a sense that it does not have an ability of adjusting the rule base in response to the change in control environments. This requires a significant calibration effort for various work environments, thus necessitating change of the fuzzy rule base, adjustment of the membership function or change of scaling factor to determine the control input. In an effort to overcome this drawback, a self organizing control(SOC) algorithm has been introduced by Procyk and Mamdani[5]. In the fuzzy SOC mechanism the controller can adapt itself to new control environments due to its self-learning ability. The algorithm has been applied to the control of a two degree of freedom robot arm[6]. More recently, several other researchers[7],[8] have worked on improvement of the SOC algorithm to reduce computational burden which makes it difficult to run with very small sampling times. Guely and Todo[7] have applied the improved algorithm to the control of two degree of freedom parallelogram type robot arm and achieved better control performance than that obtained by an optimized PID controller.

Although the previous results show that the fuzzy SOC exhibits a performance which is comparable to and often superior to a conventional PID controller, the precision of the control performance appears to be limited by less input information due to the quantization of the input error. Furthermore, there remains another problem in that the present fuzzy SOC algorithm may have undesirable control performance until the rule base converges to a stable state. In this paper, to avoid these drawbacks some modifications have been made to the SOC control system, which can be effectively implemented to the robot motion control. To investigate the performance of the proposed controller, a series of experiments are conducted for the first three joints movement of a PUMA-760 robot manipulator, whose controller is redesigned for our purpose. The basic concept and design method of the proposed control structure are described in chapter 2 and experimental results and discussions are expressed in chapter 3.

2. THE PROPOSED CONTROL STRUCTURE

Although the fuzzy SOC scheme[2] has been proved to be efficient in that it has a capability of adapting itself to varying control environments, it has some limits to be directly applied to robot control problem. First, because all the fuzzy subsets are defined on the discrete universe of discourse and the fuzzy inference is performed over such fuzzy sets, the control input is also determined by quantized value. Therefore, in spite of its effect insensitive to noise, smooth motion cannot be expected in case of the motion control of fast dynamic systems. Secondly, the procedure of the fuzzy inference which calculates the overall relationship matrix requires so much computational time proportional to the number of rules. Therefore, it causes a real time implementation problem in case of handling so many rules. Furthermore, since the number of rules is combinations of fuzzy subsets, the number of fuzzy subsets is usually limited due to the computing speed of a CPU. Otherwise, a dedicated electronic device for high speed fuzzy reasoning is needed. Finally, because the stability of the fuzzy SOC algorithm is not guaranteed until the rule base converges to a stable state, it may cause undesirable control action and safety problems when directly applied to a robot control system.

To avoid these criticisms a fuzzy SOC plus P–control structure is proposed and the fuzzy SOC algorithm introduced in [5] was modified so that it can be effectively implemented to the robot motion control.

2.1 Addition of a conventional P–control law

The proposed control system block diagram of this scheme is shown in Fig.1. The proportional control algorithm is added in parallel with the fuzzy SOC scheme for improving the convergence of learning rules and for guaranteeing the stability during intermediate operation with incomplete rule base. The merit of this combined structure is that the P–controller stabilizes the system when the fuzzy controller does not act properly, while the fuzzy SOC provides the adaptivity which the P–controller does not have.

2.2 The fuzzy SOC algorithm

As seen from Fig.1, the fuzzy SOC consists of four parts; namely the fuzzy inference mechanism, the rule base, the rule modifier, and the performance measure table. The fuzzy inference mechanism executes aggregation of control rules (MIN/MAX operations) using

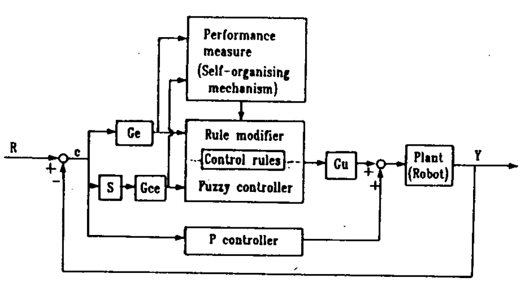


Fig.1 Block diagram of the proposed control structure

the rule base and defuzzifies the control input by the center of gravity method. The rule modifier updates the contents of the rule base according to the performance measure table.

Definition of fuzzy subsets

If the position error(E), the change in position error(CE) and the control input(U) are the basic variables obtained by scaling actual values, they are defined by

$$e(nT) = y(nT) - r(nT) \quad (1)$$
$$E = G_e \times e(nT) \quad (2)$$
$$CE = G_{ce} \times (e(nT) - e(nT-T)) \quad (3)$$
$$U = G_u \times u(nT) \quad (4)$$

where $y(nT)$ and $r(nT)$ are the measured position and the reference position of a joint at the n–th sampling time, respectively, and G_e, G_{ce} and G_u are the scaling factors which convert actual values into elements of universe of discourses. It is noted that quantization of the basic variables is not made to obtain the equations (1) through (3). This means that all the fuzzy subsets are defined in the universe of discourse(E, CE, U) which is assumed to be continuous. For E, CE and U, seven fuzzy subsets are defined having a center value of integer from -3 to 3 as shown in Fig.2. They are named as follows;

NL: negative large
NM: negative medium
NS: negative small
ZO: zero
PS: positive small
PM: positive medium
PL: positive large.

The advantage of using the fuzzy subsets defined in the continuous universe of discourse is that the fuzzy controller can generate smoother control signal than the quantized control signal.

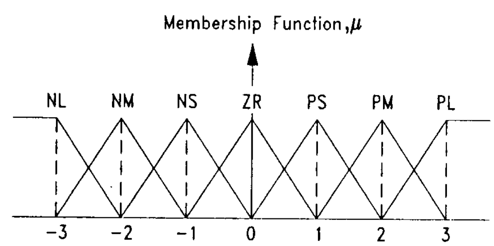


Fig.2 Membership grade function

Rule base

The rule base consists of rules which connect the error (E) and change in error(CE) with the control input(U) that has to be applied, and contains linguistic rules of the form;

IF E is E_k and CE is CE_k THEN U is U_k.

where E_k, CE_k and U_k ($k = 1$ to 7) are the fuzzy subsets defined on E, CE and U, respectively. Each non–zero entry in the rule base corresponds to one rule and the consequence of each rule in the rule base has a center value of fuzzy subset of the control input(U_k) for a pair of (E_k,CE_k). Because seven fuzzy subsets are defined respectively for E and CE, possible number of rules becomes 49 and therefore the rule base is represented by 7×7 matrix as follows;

$$U_k = Rule(E_k, CE_k) \qquad \text{for } k=1 \text{ to } 7. \qquad (6)$$

Inference mechanism

The rule base makes major role in generating the fuzzy control input. The fuzzy inference was performed in the following procedures which are repeated at every sampling period:

<STEP 1>Scaling the basic variables

From the measured position error, the normalized values, C and CE are calculated using Eq.(1) through Eq.(3). The values are ranged from -3 to 3. If they are out of the range, they are saturated to the limit.

<STEP 2>Calculation of the membership function

From the membership functions defined in Fig.2, there are only two fuzzy subsets with non-zero membership grade for a given normalized basic variable. Therefore, four membership grades ($\mu_{E_1}, \mu_{CE_1}, \mu_{E_2}, \mu_{CE_2}$) can be calculated if fuzzy subsets for E and CE, are assumed to be E_1, E_2, CE_1 and CE_2, respectively, .

<STEP 3> *Aggregation of rules*

As a result, the following 4 rules can be candidates for aggregation of rules:

$$U_1 = Rule(E_1, CE_1)$$
$$U_2 = Rule(E_1, CE_2)$$
$$U_3 = Rule(E_2, CE_1)$$
$$U_4 = Rule(E_2, CE_2).$$

Based on the Min-Max algorithm which is well known in the fuzzy control, the fuzzy inference is performed using the following operations:

(Minimum operation): If $\mu_k(U)$ is the membership grade function for the fuzzy subset of control input U which is the result of the kth rule, it can be obtained by

$$\mu_k(U) = Min(\mu_{E_i}, \mu_{CE_i}, \mu_{U_k}(U)) \qquad (7)$$

where $i,j = 1$ to 2 and $k=1$ to 4.

(Maximum operation): From the results of 4 different rules, the final membership grade function $\mu_U(U)$ can be determined using maximum operator,

$$\mu_U(U) = \text{Max}(\mu_1(U), \mu_2(U), \mu_3(U), \mu_4(U)). \qquad (8)$$

(Defuzzification): Finally, to determine the crisp value of the control input U, the defuzzification is processed using the center of gravity method,

$$U = (\sum_i \mu_U(U_i) \times U_i)) / \sum_i \mu_U(U_i) \qquad i = 1 \text{ to } 4 \qquad (9)$$

where U_i ($i = 1$ to 4) are center values of the fuzzy subsets of the control input. Then, actual control input $u(nT)$ can be obtained using Eq.(4) and Eq.(9).

Rule modification

The rule base are initially set zero and adjusted by the rule modifier during control operation. Since the entry in the rule base represents the action of a rule at the corresponding condition, its adjustment can be considered as rule creation or modification. Using the performance measure table illustrated in Fig.3, the rules are reinforced or remains the same, so as to get better control performance than the one obtained in the previous step. The rule modification can be expressed by the following two steps which are repeated at every sampling period:

<Step 1> Measurement of control performance

The scaled basic variables(E, CE) are used for the fuzzy inference and also used for measuring the control performance. The performance measuring is made by looking up the performance measure table for a pair of fuzzy subsets(E,CE). Then, the correction value $P(nT)$ can be obtained by

$$P(nT) = IM((E(nT), CE(nT)) \qquad (10)$$

where IM denotes the performance measure table or improving matrix.

<Step 2> Modification of rule

$P(nT)$ is now a quantity by which a specified rule is modified. If the system to be controlled has time delay m, $E(nT)$ and $CE(nT)$ are the result of the control input $U(nT-mT)$ generated at m samples in the past. Therefore, the rule which is most responsible for $U(nT-mT)$ should be modified. This can be described by the following equation;

$$Rule(E(nT-mT), CE(nT-mT))$$
$$= Rule(E(nT-mT), CE(nT-mT)) + P(nT). \qquad (11)$$

In the the PUMA system, m is assumed to be 1.

Error (E)	Change of error (CE)						
	NL	NM	NS	ZR	PS	PM	PL
NL	3	3	3	2	0	0	0
NM	3	2	2	2	1	0	0
NS	2	1	1	1	0	0	0
ZR	1	0	0	0	0	0	-1
PS	0	0	0	-1	-1	-1	-2
PM	0	0	-1	-2	-2	-2	-3
PL	0	0	0	-2	-3	-3	-3

Linguistic sets number of control input

PL = 3
PM = 2
PS = 1
ZR = 0
NS = -1
NM = -2
NL = -3

Fig.3 Performance measure table

3. EXPEXRIMENTS

3.1 Experimental setup

The PUMA-760 manipulator was used for the experimental test model. For the experiment, servo axis control boards were newly designed using INTEL 8032 8-bit micro controller for interfacing with a personal computer (Fig.4). The fuzzy SOC algorithm was implemented to an IBM-AT/386 personal computer which communicates with the axis control boards through parallel I/O interfaces. The axis control board executes the proportional control loop including encoder signal processing and D/A conversion of the control signal. Since the P-control action is executed in the axis control board independently from the SOC algorithm, the computational burden for adding the P-control loop can be reduced. The computation time of the SOC algorithm is about 16 milliseconds which is sufficiently small for sampling time of a servo controller.

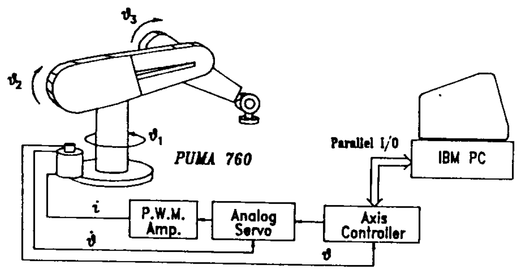


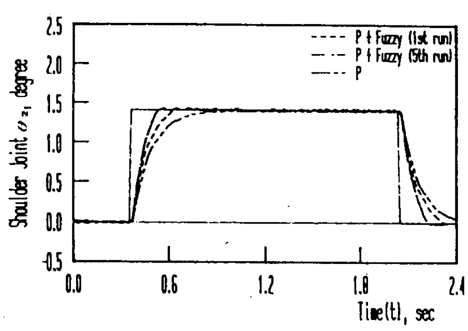
Fig.4 Schematic diagram of the experimental euipment

3.2 Results and discussions

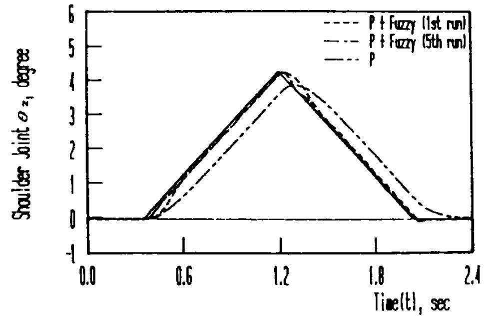
The experiments were conducted for the motion control of the first 3 joints of the manipulator for given reference trajectories. The test trajectories were given in the joint coordinates and they are step, ramp and parabolic functions.

Responses for various trajectories

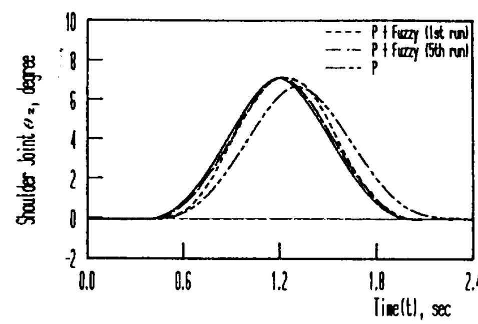
The responses of the shoulder arm for various reference trajectories, are depicted in Figs.5(a)-(c). In the figures the responses obtained by the proposed SOC algorithm are compared with those of the P-control only. They shows that the SOC is superior to P-control law: The responses are much faster and yield more accurate tracking results. It can be also observed that much better responses can be obtained as the number of iterations increases.



(a) Step response: $G_e=0.1, G_{ce}=0.15, G_u=4, K_P=1$



(b) Ramp response: $G_e=0.03, G_{ce}=0.1, G_u=4, K_P=1$

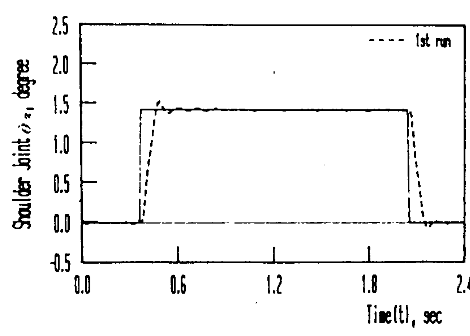


(c) Parabolic response: $G_e=0.02, G_{ce}=0,01, G_u=4, K_P=1$

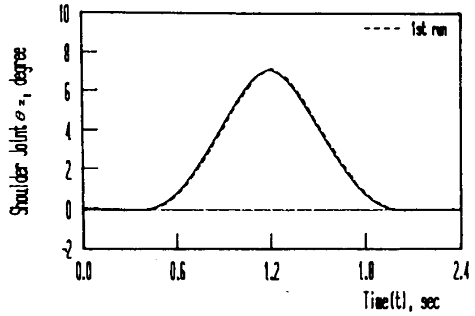
Fig.5 Experimental results of SOC without initial rules

Effect of initial rules

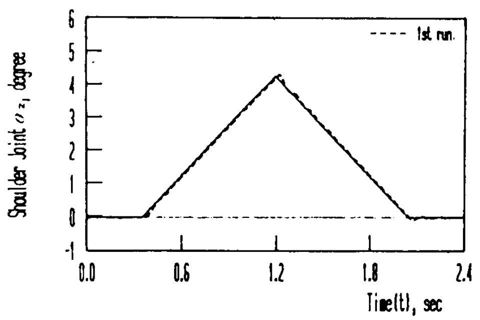
In the previous experiments, it is assumed that the rule base at initial state has zero entries. It is, however, expected that if the rule base starts with proper rules, the performance of the SOC will be improved in the view of the convergency. Figs.6(a)-(c) show that when the initial rule base starts with the final rules created during the previous control experiments conducted to obtain the results of Fig.5, the SOC converges within only one iteration. Fig.6(d) shows the tracking result for complex arbitrary input trajectory combined with several different



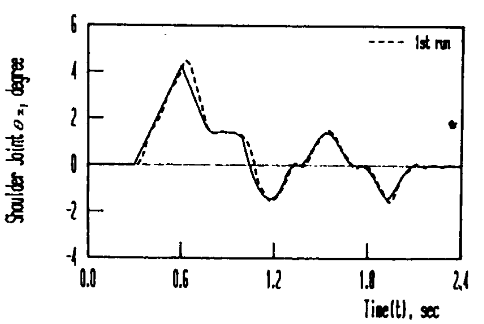
(a) Step response: $G_e=0.1, G_{ce}=0.15, G_u=4, K_P=1$



(b) Ramp response: $G_e=0.03, G_{ce}=0.1, G_u=4, K_P=1$



(c) Parabolic response: $G_e=0.02, G_{ce}=0.01, G_u=4, K_P=1$



(d) complex response: $G_e=0.02, G_{ce}=0.01, G_u=4, K_P=1$

Fig.6 Experimental results of SOC with initial rules

functions. From all these results, it can be concluded that the rule base learned from a specific input function can add learning capability when the SOC system is subjected to follow other type of input function. This indicates that the fuzzy SOC with the P–controller has fast learning convergence and good adaptivity to different types of input trajectory.

Effect of scaling factors

To investigate the effects of the scaling factors, the experiments were repeated for various scaling factors. When the performance index J is defined as

$$J=\Sigma|\,e(nT)| \quad \text{for } n=1 \text{ to } 200, \tag{12}$$

the variations of the performance index with scaling factors are depicted in Figs.7(a)–(c). It is concluded that the scaling factors greatly influence the performance. Finally the investigation on the effect of the disturbance(payload variation) was also made but its results have not been shown because in case of the PUMA robot, the difference of the performances due to payload variation is not so remarkable even for the P–control only.

4. CONCLUSION

An online fuzzy SOC method which can be run by a personal computer and its new control structure combined with conventional P–controller were proposed. The method has such features as small computation time for inference and fast convergence to reach the desired control performance. Its effectiveness was tested by applying it to the tracking control of a PUMA 760 manipulator. It is believed that the proposed control scheme and structure would be effectively applicable to the control of the plant which often faces with varying circumstances. At present, the design of feedforward controller based on this fuzzy SOC approach is of interest and its performance is being investigated.

5. REFERENCES

[1] K.S. Rattan and B. Chiu,"Rule based Fuzzy Control of a Single–link Flexible Manipulator in the Presence of Joint Friction and Load Change", *ACC*, 1989 ,pp 2749–2750.

[2] Y.F.Li and C.C. Lau, "Development of Fuzzy Algorithms for Servo systems", *IEEE Control Magazine*, April 1989.

[3] G. Vachtsevanos and H. Hexmoor, "A Fuzzy Logic Approach to Robotic Path Planning with Obstacle Avoidance", *Proc. of 25th Conf. on Dec. and Con.*,1986, pp 1262–1264.

[4] M.Sugeno and M.Nishida, "Fuzzy Control of Model Car", *Fuzzy sets and Systems*, Vol.16, 1985,pp 103–113.

[5] T.J.Procyk and E.H. Mamdani, " A Linguistic Self–organizing Process Controller", *Automatica* Vol.15, 1979.

[6] E.M. Scharf and Mandic, "The Application of a Fuzzy Controller to the Control of Multi–degree of Freedom Robot Arm", *Industrial Applications of Fuzzy Control*, 1985.

[7] F.Guely and I. Todo, "A New Fuzzy Self–organizing Control Algorithm and its Application to the Control of a Robot Arm, *Proceedings of 20th ISIR*, 1989.

[8] S.Daley and K.F.Gill, "Comparison of a Fuzzy Logic Controller with a P+D Control law", *ASME J. of Dyn. Sys., Mea. and Con.*, 1986, pp 128–137.

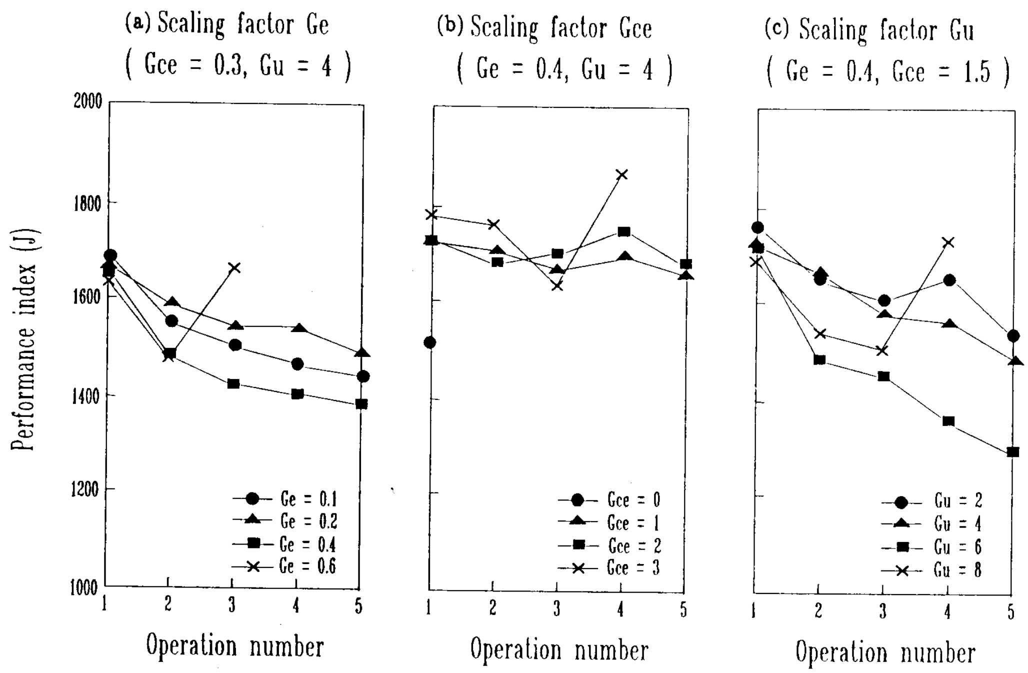


Fig.7 Variation of performance with scaling factors