Design of Fuzzy PID Controllers Using Steady-state Genetic Algorithms

권 영섭, 샤오 웬동
Yeong-Seob Kueon and Wendong Xiao
포항제철기술연구소, 계측제어연구팀
Email: pc548332@smail.posco.co.kr

Abstract

In this paper the steady-state genetic algorithm is applied for the optimal design of fuzzy PID controllers. Basically the structure of the discussed fuzzy PID controller is extended from the conventional fuzzy PI and PD controllers where only a two-dimensional rule base is used. By SSGA, the fuzzy membership functions, scaling factors and rule base of the fuzzy PID controller are designed simultaneously. Simulations results shows the superior performance of this optimal designed fuzzy PID controllers to the optimal designed conventional fuzzy PI and PD controllers.

Keywords: fuzzy logic control, genetic algorithm, steady-state genetic algorithm

1. Introduction

Fuzzy logic control(FLC) has received increasing interest in the control community, and many progresses in its theoretic studies and practical applications have been reported. The main merits of FLC is that it can express and implement linguistic control strategy of expert knowledge without a precise mathematical model of the controlled system[1, 2, 3].

FLC is usually used as proportional-integral type (fuzzy PI) or proportional-derivative type (fuzzy PD), which uses error and error derivative as inputs and respectively increment control signal or control signal directly as output. Fuzzy PID controller is also generally used by generating incremental control output from error, error derivative and acceleration error. Usually fuzzy PID controller can get superior performance than fuzzy PI and PD controllers, however, the more number of inputs will expand rule base greatly and makes the design task more difficult[4, 7].

Recently a new fuzzy control structure is presented and analysed by several authors[4, 5, 7, 8], it is extented from the conventional fuzzy PI and PD controllers. It only uses error and error derivative as inputs, and includes integration and proportional action for output of the inference output, so only two-dimensional rule base is required. It is simple in structure, and easy for implementation, more important, it can achieve excellent control performance with response to the transient and steady-state response, robust to the internal and external disturbances[4,7].

However, owing to the large number of parameters contained in the afore-mentioned fuzzy PID controller, which consists of determinitation of fuzzy membership functions(MFs), scaling factors(SFs) and rule base, to obtain an optimal fuzzy PID controller is not an easy work. Some attempts has been done, In [7], the authors use a heuristic tuning method by two level tuning strategy based on the standard MFs and a general rule base. Paper [6] presents a two phase method for the optimal design of MFs and scaling factors including a first rough tuning phase of SFs by means of Rosenbrok identification method and a second phase for fine tuning of Mfs and Sfs by the gradient descent method. However, their results can not asure to find the optimal solutions and rely on the predefined fuzzy rule sets.

Recently more and more works has suggested the genetic algorithm[9, 10, 11] as an effective and efficient method for the optimal design of FLC. For example, C. Karr used GA to derive the membership functions for predefined sets of rules[12], Lim et al. a GA paradigm for learning fuzzy rules[13], Linkens et al.[14] and Homaifar et al.[15] investigated how to design the membership functions and the rule base together.

In this paper we will develop the technique for the design of afore-mentioned fuzzy PID controllers by means of GA, we use the steady-state genetic algorithm(SSGA), in which a steady-state reproduction scheme is adopted. The fuzzy membership functions, scaling factors and rule base of the fuzzy PID controller are designed simultaneously.

The paper is organized as follows, first, the basic concepts of fuzzy PID controller and genetic algorithms are briefly introduced in Section 2 and 3, the implementation details of the steady-state GA for the fuzzy PID controllers are described in section 4, comparison results of the optimal designed fuzzy PID controllers with the optimal designed conventional fuzzy PI and PD controllers are shown in Section 5, and concluding remarks are given in Section 6.

2. PI, PD and PID type Fuzzy Controllers

FLC is usually used as proportional-integral type (fuzzy PI) or proportional-derivative type (fuzzy PD), which uses error and error derivative as inputs and respectively increment control signal or control signal directly as output. As reported in [7], fuzzy PI controller has good performance at steady state (without steady-state error), but gives poor performance in transient response with long rise time and a long settling time due to the internal integration operation. On the other hand, fuzzy PD controller has good performance at the transient state, but will cause the steady-state error or the steady-state oscillation. Fuzzy PID controller is also used usually by generating incremental control output from error, error derivative and acceleration error. Although it can enhance the performance a lot, however, the more number of inputs will expand rule base greatly and makes the design task more difficult.

The new fuzzy PID controller structure is shown in Fig. 1, clearly, it only uses error and error derivative as inputs, and includes integration and proportional action for output of the inference output, therefore only a two-dimensional rule base is required. The scaling factors

Ku_1/Ku_2 stand for the contributions of the integration and derivative operations. Easy to see, the conventional fuzzy PI controller and PD controller are special cases of this control structure corresponding to the output scaling gains $Ku_1=0$ and $Ku_2=0$ respectively.

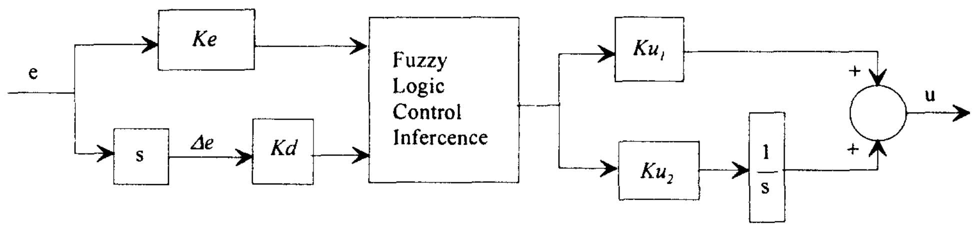

Fig. 1: The Structure of fuzzy PID Controller

The fuzzy inference process consists of fuzzification, decision making logic and defuzzification process. Inputs of error and error derivate are normalized by scaling factors, the fuzzification action enables to define a grade of membership to fuzzy sets characterizing these two normalized input variables. The rule base are considered by the fuzzy decision making logic. The result of the defuzzification is incrementally processed by proportional and integral actions weighted by the two scaling factors Ku_1 and Ku_2 and obtain the control output to the controlled system.

In this paper, seven fuzzy sets are assigned for each input/out fuzzy variables, i.e. ngative large(NL), negative medium(NM), negative small(NS), zero(ZE), positive small(PS), positive medium(PM), and positive large(PL). Triangular-shaped membership functions are used for fuzzy sets of each input, singleton-shaped membership functions are used for those of the output, their universes of discourse are all normalized in the interval [-1, 1]. A bezdeks repartition[5] is satisfied for each fuzzy sets of the input variables, i.e. the maximum of a membership function(corresponding to the peak of the triangular) equals to the minimum of the adjacent membership functions. The well known Min-Max method and the weighted average method are used for fuzzy inference and defuzzification.

3. Genetic algorithms and steady-state genetic algorithm

Genetic algorithms are a class of global search and optimization procedures based on the mechanics of natural selection and genetics[9, 10, 11]. The algorithm starts from an initial set of solutions and uses a process similar to the biological evolution to improve upon them. Because GAs avoid many of the shortcoming exhibited by local search techniques on difficult search spaces such as not relying on the gradient information and parallel, stochastic natures, they have been successfully used in various areas.

GAs encode each candicate solution by chromosome, usually used as binary bit string, the fitness of each chromosome is evaluated as the qualititative measure of how well it can represent a solution of the problem. The algorithms work on a population of the chromosomes over a sequence of generations using the applications of the reproduction, crossover and mutation operations.

Reproductive is done by repeatedly selecting members from the old population, according to their fitness values, and adding these members to the new population. Selection is biased by fitness, where fitter chromosomes are given higher probabilities, so they can produce more copies in the new population. There are two reproductive techniques in general use, the generational reproduction and the steady state reproduction. Briefly, generational reproduction replaces the entire population at once, while steady-state reproduction replaces one or two members at a time. Tere have been some empirical evidences which demonstrate the advantages of SSGA, The main argument for SSGA is that the new offspring can participate in information exchange with other chromosomes as soon as they are produced.

Crossover operates on two newly reproduced chromosomes (parents) at a time and generates offsprings by exchanging parts of the structures of the two chromosomes, resulting in exchange of information. Mutation operates on a single chromosome and generates offspring by altering one or more genes. usually randomly applied at a large probability,, usually randomly applied at a lower level bit probability,

The SSGA is the GA used in this paper. Roulette wheel is used as the selection mechanism. Two parents are first selected from the population and producing two offsprings by crossover and mutation operations, the one of the offsprings with lower performance is discarded, the other one replaces the worst individual in the population.

4. SSGA for fuzzy PID controller

The design problem of the fuzzy PID controller includes tge determination of the input and output scaling factors, the design of membership functions, and the generation of the fuzzy rule base. The detailed design scheme of SSGA will be described in this section.

4.1 Coding of the solution

Binary coding is used in this paper. Each chromosome of the fuzzy controller structure consists 3 segments corresponding to the input and output scaling factors, membership functions, the rule base. They are shown in Fig. 2.

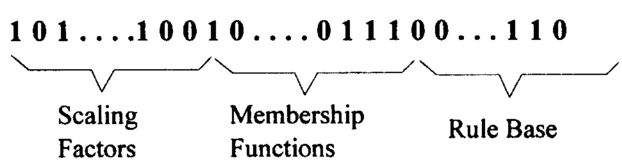

Fig.2 coding of the fuzzy controller

The first segment represents the scaling factors Ke, Kd, Ku1 and Ku2, there are 10 bits for each of these values, so 40 bits are required.

The second segments represents the membership functions. For each input variable, only the peak values corresponding to NL, NM, NS, PS, PM, PL are defined, each one coded by 6 bits. The peak value of fuzzy set ZE is assumed 0 and not included in the coding. Similarly, for the fuzzy output variable, the singleton value of NL, NM, NS, PS, PM and PL are coded, each one by 6 bits. The singleton value of fuzzy set ZE is also assumed 0 and not included in the coding. Therefor 108 bits are included in the second segment.

The third segment are the coding of the rule base. Since the controller should operater in a stable region, four different areas are clarified as shown in Fig. 3:

1. In the areas denoted by the signal −, the fuzzy set of the rule consequent belongs to { NL, NM, NS, NULL} where NULL denotes no rule applied, they are coded by 2 bits, with 00 denotes NL, 01 denotes NM, 10 denotes NS and 11 denotes NULL;
2. In the areas denoted by the signal +, the fuzzy set of the rule consequent belongs to {PS, PM, PL, NULL}, they are also coded by 2 bits, with 00 denotes PS, 01 denotes PM, 10 denotes PL and 11 denotes NULL;
3. In the areas denoted by the signal *, the fuzzy set of the rule consequent belongs to {NL , NM, NS, ZE, PS, PM, PL, NULL} and are coded by 3 bits;
4. In the area denoted by the signal 0, the fuzzy set of the rule consequent belongs to {ZE, NULL} and is coded by 3 bits;

Therefore, in segment 3, 103 bits are required to code the rule base.

E \ SE	NL	NM	NS	ZE	PS	PM	PL
NL	−	−	−	−	−	−	*
NM	−	−	−	−	−	*	+
NS	−	−	−	−	*	+	+
ZE	−	−	−	0	+	+	+
PS	−	−	*	+	+	+	+
PM	−	*	+	+	+	+	+
PL	*	+	+	+	+	+	+

"−" ∈{NS, NM, NL, Null}; "+" ∈{PS, PM, PL, Null}
"*" ∈{NS, NM, NL, ZE, PS, PM, PL, Null}; "0" ∈{ZE, Null}; Null=No Rule

Fig. 3: Representation of the rule base

4.2. Performance Criteria and fitness technique

The performance is measured based on the following criteria:

1. overshoot(OV);
2. rise time(ST) corresponding to the time it take to the 90% of the set-point;
3. Integral of time absolute error(ITAE) $\int t|e(t)|dt$.

Overshoot and rise time are used to measure of the system transient response. ITAE keeps account of the errors at the beginning and also emphasises the steady state.

The overall performance is defined as

$$f = RT * ITAE * (1 + k * OV) \quad (1)$$

where k is a parameter of the performance. The fitness of a chromosome(controller parameter) is first transfered to f by formulae (2) and then processed by the linear scaling technique

$$f' = f'_{min} + (f_{max} - f) / (f_{max} - f_{min}) \quad (2)$$

where f_{min}, f_{max} are the minimal and maximal overall performance, f' is the minimal value for the fitness f.

4.3. SSGA parameter

The parameters for the SSGA are set as follows:

1. Population size: 50;
2. Maximal number of generation: 2000;
3. Crossover: uniform crossover with probability 0.8;
4. Bit mutation probability: 0.004.

5. Simulation results

As an example, a third-order linear model[16] is chosen as

$$\frac{1}{s^3 + 1.75s^2 + 2.15s + 1} . \quad (3)$$

The numerical integration method used is the 4-th order Runge-Kutta method, the integration interval is chosen as 0.1s.

Comparison is given between the SSGA optimal designed fuzzy PID controller and the optimal designed conventional fuzzy PI and PD controllers with respond to the transient and steady-state response. The SSGA for fuzzy PI and PD controller is similar to the design of the fuzzy PID controller, the differences are only at the coding scheme of the first segment. For fuzzy PD controller, only the *Ke, Kd* and *Ku1* are encoded, each one by 10 bits. For fuzzy PI controller, only the *Ke, Kd* and *Ku2* are encoded, each one also by 10 bits.

The performance of the SSGAs for the 3 fuzzy controller is shown in Fig. 4, which demostrates the current best in the population averaged over 20 runs, From Fig. 4, it is clear that the overall performance of the SSGA for fuzzy PID controller outperform the other two conventional fuzzy controllers, with the fast converge speed, which means it is easy for SSGA to find a good control output within small time.

The statistics results are further shown in Table 1 for the average best solution over 20 runs and the best solution found in 20 runs, Fig. 5 gives examples of the SSGA running results for the fuzzy PI, PD and PID controllers. It shows the advantages and disadvantages of the 3 kinds of FLC structures.

Here fuzzy PI controller have worst overall performance, it can have small overshoot, but the rise time and ITAE are large, they are caused by the slow speed of the integration operation. As shown in Fig. 5, the advantage of fuzzy PI controller is that it has good performance at steady state without steady-state error.

Here fuzzy PD controller have worse overall performance, its rise time is smallest, but it has large overshoot and large ITAE. As shown in Fig.4, fuzzy PD controller can achieve fast transient response but the steady-state response is worst because it can not eliminate steady-state error.

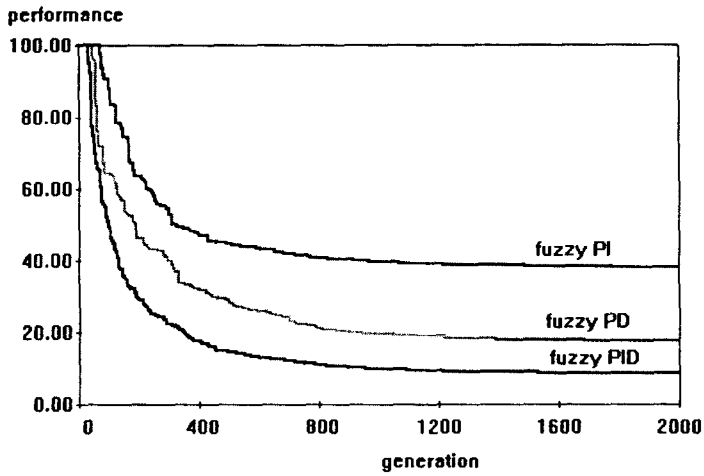

Fig. 4. SSGAs for fuzzy PI, PD, and PID controllers

	Overall Performace	Overshoot	Rise Time	ITAE	Overall Performace	Overshoot	Rise Time	ITAE
Fuzzy PI	38.2073	0.0093	4.9500	6.7158	23.5461	0.0179	3.9000	4.7613
Fuzzy PD	17.7185	0.0581	2.1500	4.4264	10.3425	0.0433	1.800	3.4828
Fuzzy PID	9.2231	0.0492	2.3700	2.1607	4.9370	0.0291	1.900	1.8086
	Average best solution over 20 runs				Best solution found in 20 runs			

Tab

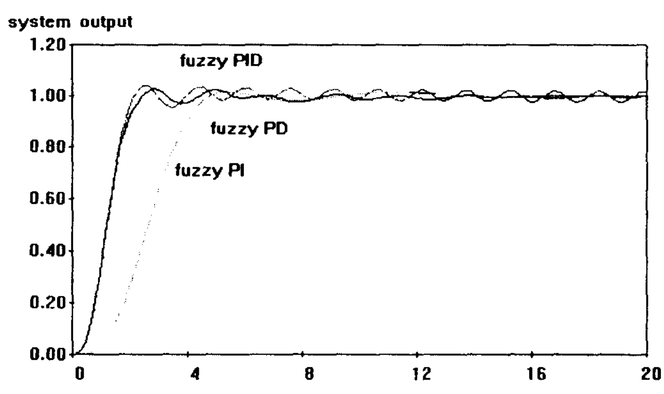

Fig. 5: Example results of the fuzzy PI, PD, and PID Controllers after SSGA

Fig. 6 and Table 2 demonstrates an example of the fuzzy membership functions and rule base found by the SSGA. The scaling factors are $Ke=0.4985$, $Kd=0.5347$, $Ku1=1.5445$, $Ku2=9.4819$.

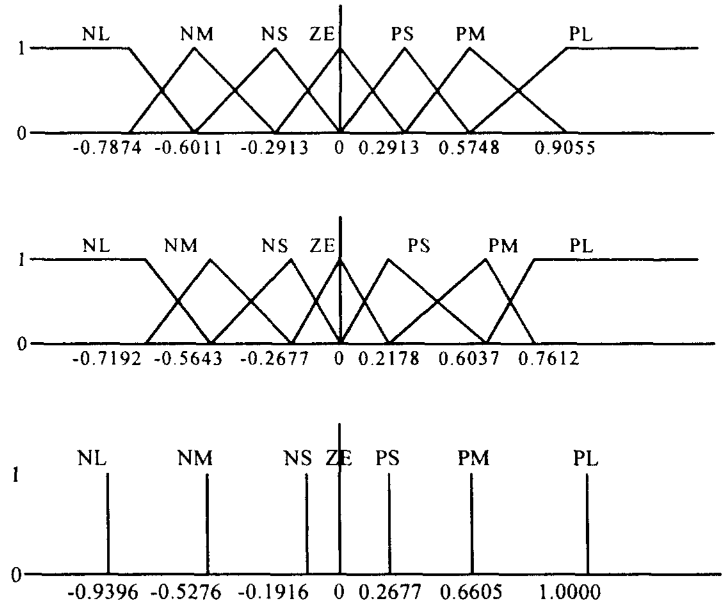

Fig 6: Fuzzy sets for error, change of error, and output found by SSGA

E\SE	NL	NM	NS	ZE	PS	PM	PL
NL	NL	NS		NL	NL	NM	PM
NM	NS	NS	NL	NL		PS	
NS	NS				PL	PS	PM
ZE		NS		ZE	PM	PS	
PS	NS	NS	ZE	PM		PM	PL
PM	NM	PS		PS			PS
PL	NM	PS	PM		PS	PL	PS

Table 2: Fuzzy rule base for fuzzy PID found by SSGA

6. Conclusion

We proposed the steady-state genetic algorithm method for the optimal design of fuzzy PID controllers, the fuzzy membership functions, scaling factors and rule base of the fuzzy PID controller are designed simultaneously. Simulations results shows the superior performance of this optimal designed fuzzy PID controllers to the optimal designed conventional fuzzy PI and PD controllers. This research suggest the SSGA as a effect and efficient design for complex fuzzy controllers. Further research is undergoing for the applications of the genetic algorithm based fuzzy PID ontrollers, analytical properties such as stability and robustness of the fuzzy PID controller are also the important issues to be addressed.

References

1. Mamdani, M. (1974), Application of fuzzy algorithms for control of simple dynamic plant, Proc. IEE, Vol. 121, Dec, pp. 1585-1588.
2. Lee, C. C. (1990), Fuzzy logic in control systems: fuzzy logic controllers----Parts I & II, IEEE Trans. Syst. Man Cybern., Vol. 20, No. 1, pp. 404-435.
3. Yen, J., R. Langari, and L. A. Zadeh(eds.) (1994), Industrial applications of fuzzy logic and intelligent systems, IEEE Press.
4. Xiao W. D., and Y. S. Kueon, Fuzzy mold level control for continuous steel casting, POSCO Report, Korea, Apr., 1998.
5. Boverie, S., B. Demaya, R. Ketata, and A. Titli, Performance evaluation of fuzzy controllers, IFAC Intelligent Components and Instruments for Control Applications, Malaga, Spain, 1992.
6. Demaya, B., R. Palm, S. Boverie, and A. Titli, Multilevel qualitative and numerical optimization of fuzzy controller, Proc. IEEE Internat. Conf. on Fuzzy Systems, 1995, 1149-1154.
7. Li , H. X., H. B. Gatland (1996), Conventional fuzzy control and its enhancement, IEEE Trans. Syst. Man Cybern.----Part B: Cybern., Vol. 26, No. 5, pp. 791-797.
8. Wu, Z. Q., and M. Mizumoto, PID type fuzzy controller and parameter adaptive method, Fuzzy Sets and Systems, 78(1996), 23-35.
9. Holland, J., Adaptation in natural and artificial systems, The University of Michigan, 1975.
10. Goldberg, D., Genetic Algorithms In Search, Optimization, And Machine Learning. Addison-Wesley, Reading, Mass., 1989.
11. L. Davis(ed.), Handbook of genetic algorithms, Van Nostrand Reinhold, New York, 1991
12. Karr, C., Genetic algorithms for fuzzy controllers, AI Expert, Feb., 1991, 26-33.
13. Lim, M. H., S. Rahardja, and B. H. Gwee, A GA paradigm for learning fuzzy rules, Fuzzy Sets and Systems, 82(1996), 177-186.
14. Linkens, D. A., and H. O. Nyongesa, Genetic algorithms for fuzzy control Part 1: offline system development and application, IEE Proc. Control Theory Appl., 142(3), 1995, 1621-176
15. Homaifar, A., and E. McCormick, Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms, IEEE Trans. on Fuzzy Systems, 3(2), 1995, 129-139.
16. Masten, M. K., H. E. Cohen, An advanced showcase of adaptive controller designs, International Joural of Adaptive control and Signal Processing, 1990, Vol. 4, pp. 89-98.