Optimization of Fuzzy Car Controller Using Genetic Algorithm

Bong-Gi Kim, Jin-kook Song, Chang-Doon Shin, *Member, KIMICS*

Abstract—The important problem in designing a Fuzzy Logic Controller(FLC) is generation of fuzzy control rules and it is usually the case that they are given by human experts of the problem domain. However, it is difficult to find an well-trained expert to any given problem. In this paper, I describes an application of genetic algorithm, a well-known global search algorithm to automatic generation of fuzzy control rules for FLC design. Fuzzy rules are automatically generated by evolving initially given fuzzy rules and membership functions associated fuzzy linguistic terms. Using genetic algorithm efficient fuzzy rules can be generated without any prior knowledge about the domain problem. In addition expert knowledge can be easily incorporated into rule generation for performance enhancement. We experimented genetic algorithm with a non-trivial vehicle controling problem. Our experimental results showed that genetic algorithm is efficient for designing any complex control system and the resulting system is robust.

Index Terms— Genetic algorithm, Fuzzy car controller, Optimization

I. INTRODUCTION

Fuzzy logic controller (FLC) is intended to control target system, using fuzzy control rules and membership functions. Fuzzy control rules and membership functions have usually been designed by human experts of the problem domain in issue. However, the conventional fuzzy logic design approach cannot guarantee optimization of fuzzy rules or membership functions proposed by human experts and, as it is always the case, such rules or functions are open to subjective judgments. Hence, several fuzzy logic design strategies using neural network or genetic algorithm have been proposed to optimize fuzzy rules and membership functions. This paper aims to optimize fuzzy rules and membership functions by applying genetic algorithm to simulated fuzzy car controller.

II. Research

Fuzzy logic controller refers to a system that expresses professional knowledge based on knowledge and experience of human experts and activates controller by inferring the aforementioned rule. Such a system is used when it is complex or difficult to express a system mathematically and able to infer control rules from intuition of human experts, empirical rules, test or simulation data without having to depend on complicated mathematical models. Fuzzy logic controller consists of four components including fuzzification moule, defuzzification module, fuzzy inference engine, and knowledge base as shown in Fig. 1.

① Fuzzification module converts the inputs to the controller into membership grades of fuzzy sets with the help of membership functions.

② Knowledge base comprises of a data base and fuzzy control rule base which characterizes the desired output response applied by means of a set of control rules. Knowledge base is linguistic type of IF-THEN statements involving fuzzy sets, fuzzy logic, and fuzzy inference. Knowledge base is an important component of a fuzzy controller that captures the operator knowledge about the system in the form of fuzzy rules. Developing a rule base is one of the most time consuming part of designing a fuzzy logic controller. Usually it is very difficult to transform human knowledge and experience into a rule base of fuzzy logic controller

③ Fuzzy inference engine is the execution module of fuzzy logic controller and infers consequential fuzzy sets based on fuzzified inputs, using inference synthesis rules and referring to knowledge base.

④ Defuzzification module is intended to defuzzify inferred fuzzy control values into a single value to enable values assigned to output variables to be mapped to values of the entire fuzzy set. Defuzzifying techniques utilizes center of mass technique, mean of maximum or standard deviation of maximum.

Fuzzy logic controller is a logical system designed to model human inference technique based on uncertain approximation and able to provide a supple and robust intelligence that can infer uncertain problem domain encompassing uncertainty as in human judgment. When compared to the existing logics, fuzzy logic has characteristics similar to those of human ideas or natural language and is often applied to pattern recognition or

---

Manuscript received March 5, 2008; revised May 14, 2008.

Bong-Gi Kim is with the Division of Computer Science & Engineering, Jinju National University, Jinju, Korea (Email: bgkim@jinju.ac.kr)

Jin-Kook Song is with the Division of Computer Science & Engineering, Jinju National University, Jinju, Korea. (e-mail : jksong@jinju.ac.kr)

Chang-Doon Shin is with the Department of Visual Contents, Hallym College, Korea(e-mail : cdshin@hsc.ac.kr)

system control that involves uncertain and approximate information. [1].

It has usually been the case that fuzzy rules and membership functions are designed by human experts of selected problem domain to provide inputs for fuzzy logic controller design. However, it is not easy to find a human expert who has experienced a lot of issues involved in a selected problem domain. Moreover, it often ends up with creation of insufficient and inefficient rules and waste of time and money resulting from trials and errors to design a fuzzy logic controller without human experts. Even if right human experts are found, there is no guarantee that fuzzy rules or membership functions designed by human experts are optimized as human beings are at the mercy of subjective judgment rather than objective perspective in general. Furthermore, it is difficult to redefine a fuzzy logic controller when a target system is changed. As a solution to the aforementioned issues, neural network-based fuzzy rule learning mechanism or genetic algorithm has been proposed as a new fuzzy system designing approach.

Among such design approaches, this paper proposes a fuzzy control system design using genetic algorithm. Genetic algorithm can enable controller to be designed even without knowledge of problem domain. In other words, if knowledge about selected problem domain is available, more efficient fuzzy rules can be designed more effectively even without involving human experts. In addition, flexibility of fuzzy logic controller can be enhanced so that it can respond to changes to target system environment or mechanism.

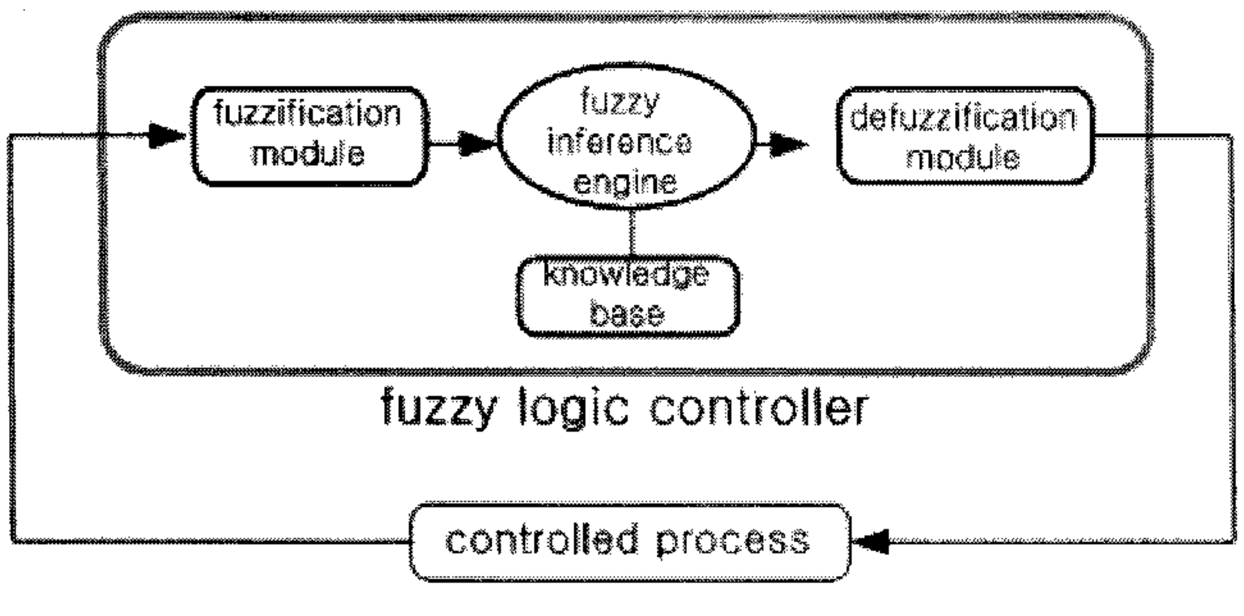


Fig. 1 Structure of Fuzzy Logic Controller

III. Fuzzy Car Controller Configuration

The system proposed herein is configured to enable fuzzy logic controller to evolve, using the framework of evolutionary algorithm as shown in Fig. 2. Firstly, parameters required for car control are defined, genes are designed, using the parameters and parameter values are initialized at random to derive a probable solution (simulated car controller). Then, car driving simulator is used to simulate a car and assess controller (genes) so that the genes can be optimized by an evolutionary loop. Fig. 3 shows the entire system configuration of fuzzy controller that uses fuzzy rules and membership function using genetic algorithm.

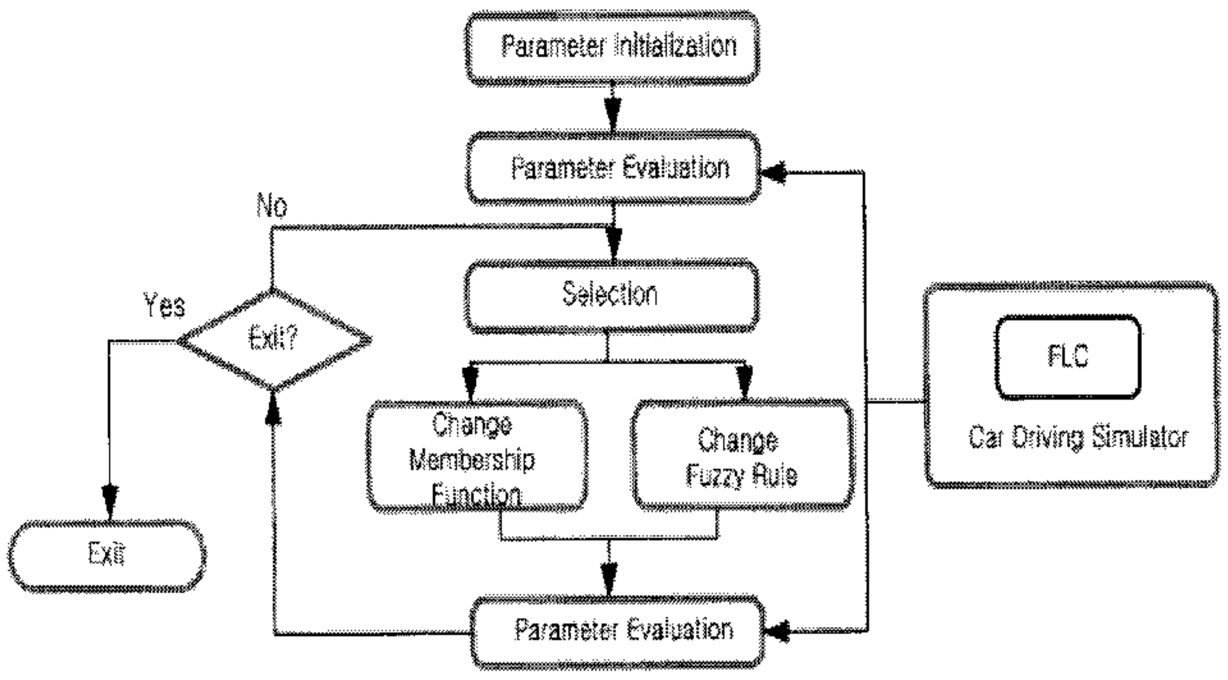


Fig. 2 Car Driving Simulation System

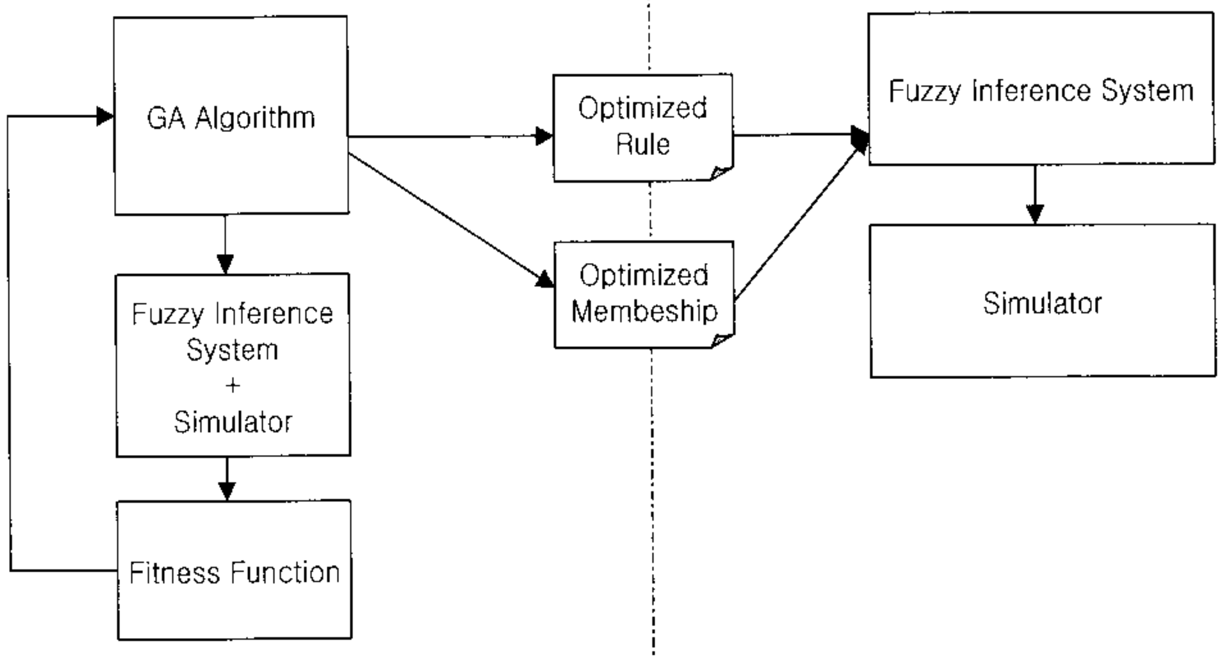


Fig. 3 Optimization Structure

IV. Car Driving Control Using Fuzzy Logic Controller

The simulated car controller proposed herein uses the fuzzy logic controller described in the above. Car and road required for car driving simulation were defined and fuzzy logic controller was used to control the simulated situation.

A. *Car Definition*

As it is difficult to simulate a real car in terms of implementation, car was defined as a dot herein and car bearing, speed, wheel alignment data were stored. As indicated in. Fig. 4, data required for car driving is stored in a data structure.

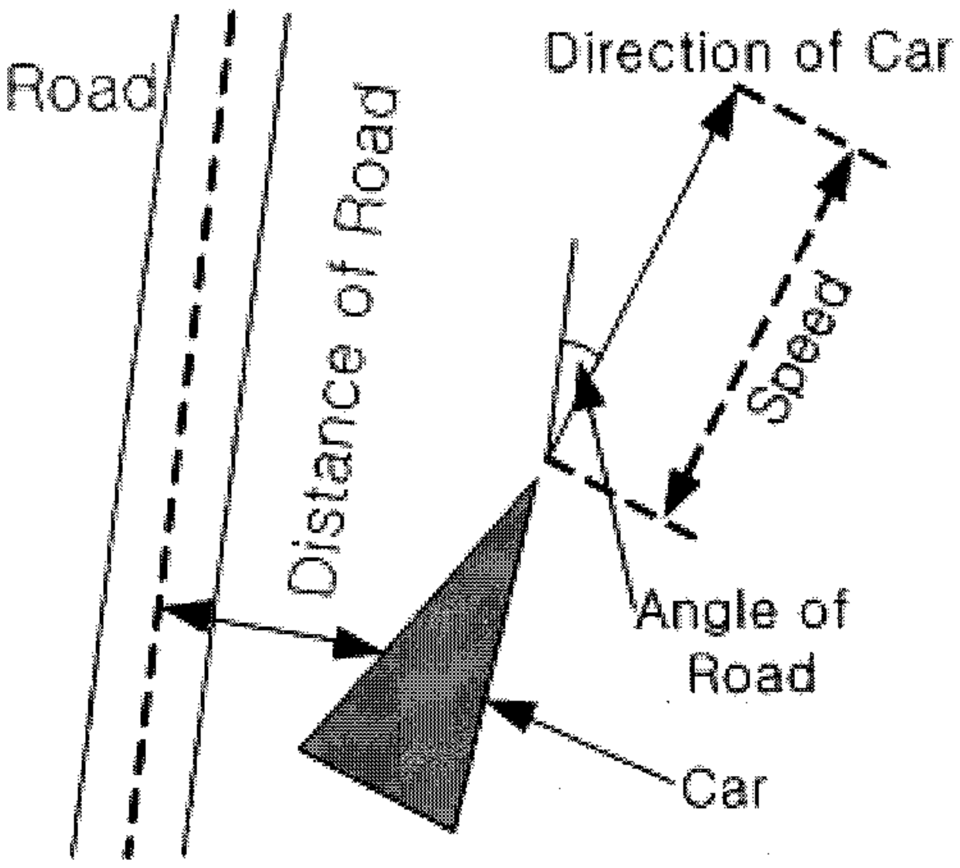


Fig. 4 Car modeling

B. *Road Definition*

Roads used herein are defined as various road alignments including straight one, curved one, rotating one and complex one. To define road data in a way that can express real road data as faithfully as possible, center points along the roads were identified and stored to express straight, curved, rotating road alignments as well as irregular one. Three elements including X axis coordinate, Y axis coordinate and the angle to the next center point were used to express a center point. The coordinates of road center point were located on the assumption that the road is located over an absolute coordinate system. When a car moves actually, only relative location of each point is significant while the absolute location is not subject to car movement and simply used to model road data. 1m was defined as the minimum unit as significant change such as abrupt change of angle is not observed in real road in 1m increment.

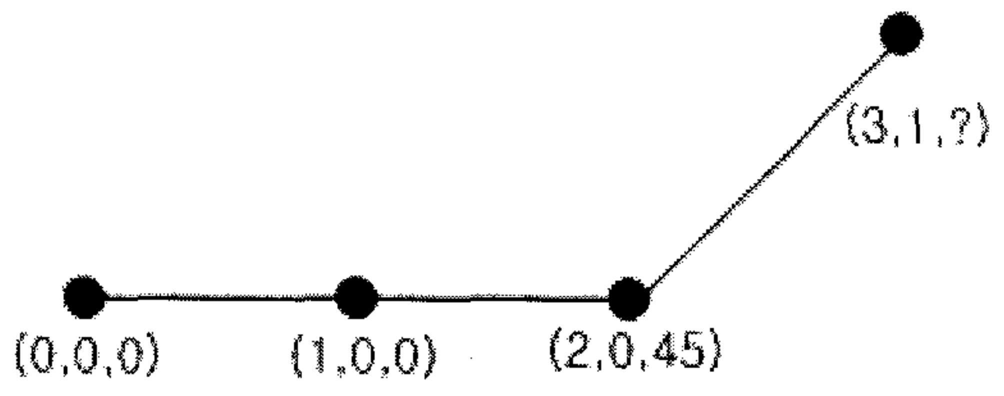


Fig. 5 Road Expression

For example, a road that goes straight and turn at 45 degrees can be expressed as (0, 0, 0; 1, 0, 0; 2, 0, 45, 3, 1, ?) in a text file. The road data defined in the above indicate that the road goes straight for 2 meters and turn at 45 degrees upward. As the angle value is not defined at the last point, it is defined as '?'.

C. *Car Driving Control*

The simulated car controller used herein is supposed to control car steering wheel (car wheels) to move the car running at variable speed along the given driving line (the center line of given road) as fast as possible. The fuzzy logic controller takes the distance and the angle between the car and the road, the car speed, the road curve as inputs from onboard sensors to output car steering angle and acceleration. Car driving on the road was controlled in a manner specified in Fig. 6.

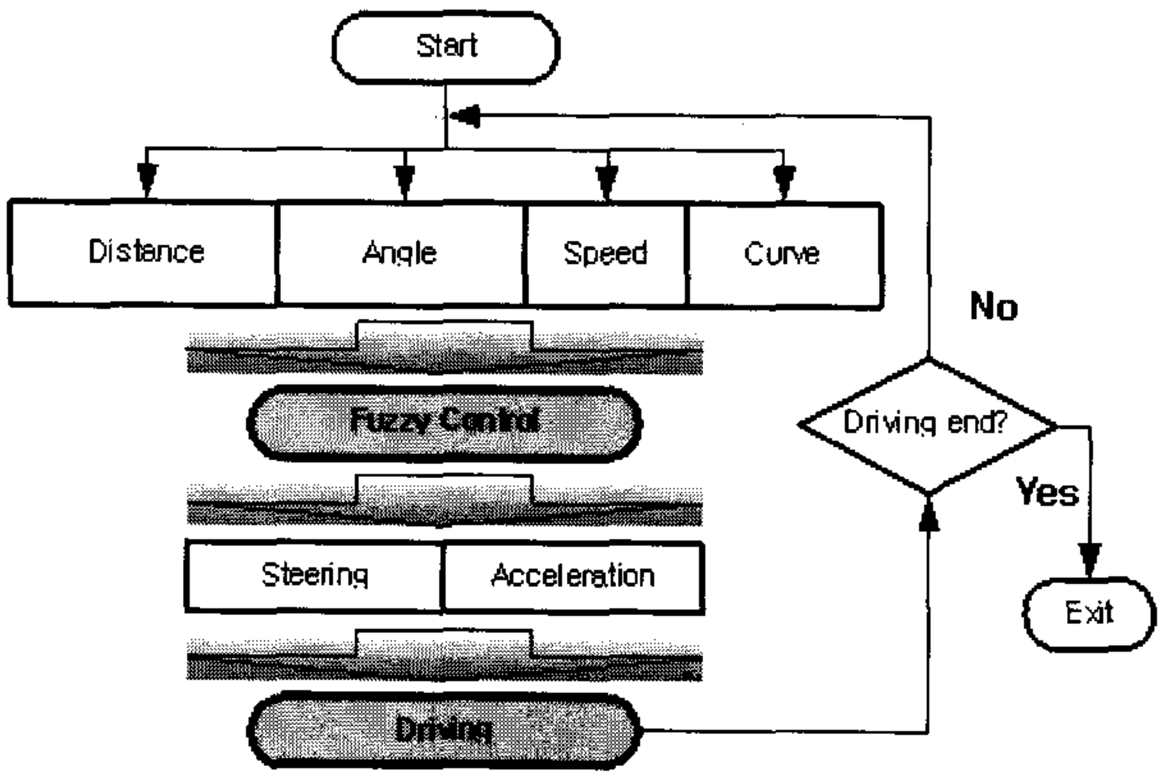
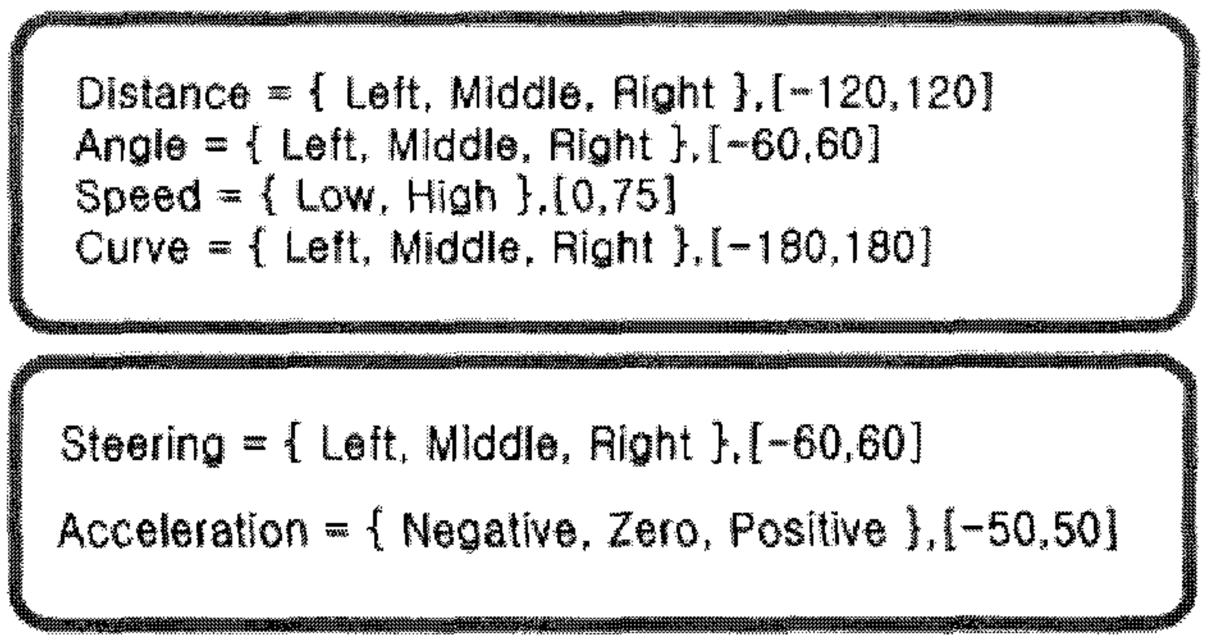


Fig. 6 Car Driving Control Model

The input variables used by the fuzzy logic controller include distance and angle between the car and the road, car speed and road curve and the output variables are car steering angle and acceleration. The assembly and the respective scopes of linguistic terms used by each variable are as described in Fig. 7.



Distance = { Left, Middle, Right },[-120,120]
Angle = { Left, Middle, Right },[-60,60]
Speed = { Low, High },[0,75]
Curve = { Left, Middle, Right },[-180,180]

Steering = { Left, Middle, Right },[-60,60]
Acceleration = { Negative, Zero, Positive },[-50,50]

Fig. 7 Input/Output Variables

In Fig. 7, the distance variable is supposed to have negative value when the car is in the left hand side of the road from the driving line and positive value when the car is in the right hand side. Supposing the driving direction on the driving line is 0, the angle, the curve and the steering variables are also expected to have negative values when the car leans to the left and positive values if it leans to the right. The unit of distance was the distance between the points on computer display and 'degree' was used as the unit of the angle, the curve and the steering variables. Supposing that unit time is defined as the time that elapses between each control action, speed was defined as the distance traveled during the unit time and acceleration as the variation of speed during the unit time.

The number of rules is dictated by the set of input variables and the fuzzy logic controller has up to 54 rules in reference to 4 input variables.

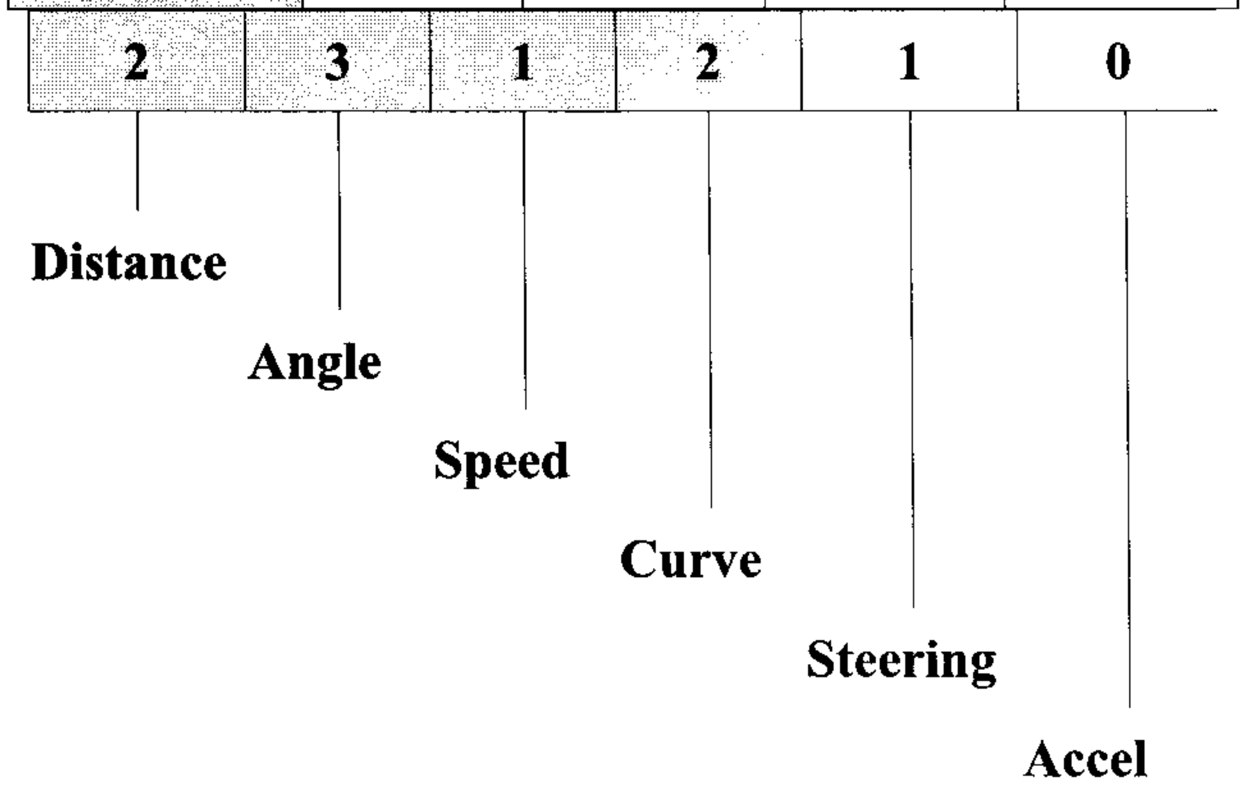
V. Application of Genetic Algorithm

A. Chromosome Composition

1) Chromosomes of Fuzzy Rules & Membership Functions

Encoding structures for the chromosomes of fuzzy logic controller are distinguished between the one for rule base and the other for membership functions. Separation of chromosome encoding structures can allow evolutionary operators to be differentiated in reference to respective characteristics of the rules and the membership functions, facilitating the maintenance of the rules and the membership functions as well as the analysis of encoded entities. Fig. 8 and Fig. 9 describe respective encoding approaches for rule base and membership functions. As for the rule base, each linguistic term is encoded as integer while it is encoded as real number in case of membership functions.

	0	1	2	3
Distance	X	Left	Middle	Right
Angle	X	Left	Middle	Right
Speed	X	Low	High	X
Curve	X	Left	Middle	Right
Steering	X	Left	Middle	Right
Accel	X	Negative	Positive	X



IF Distance is **Middle** &
Angle is **Right** &
Speed is **Low** &
Curve is **Middle**
Then
Steering is **Left**

Fig. 8 Rule Encoding Structure

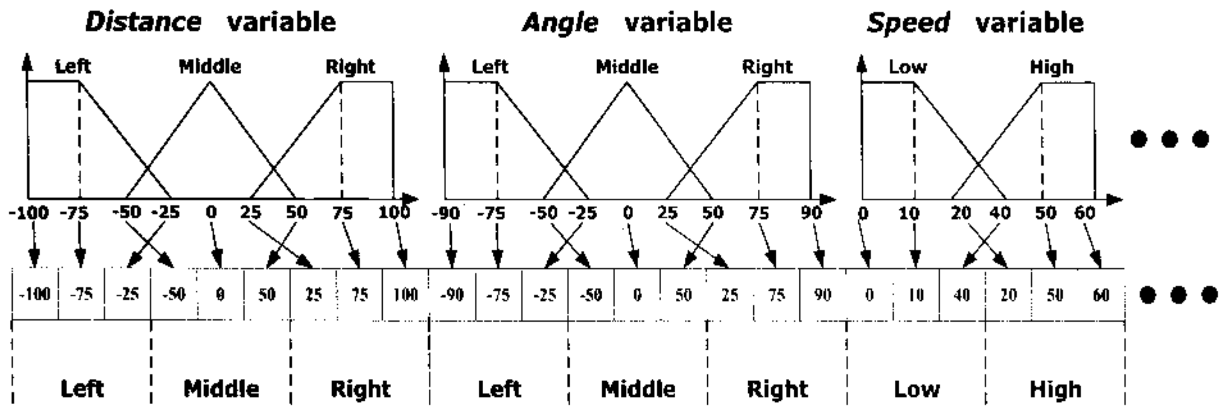


Fig. 9 Membership Function Encoding Structure

VI. Experiment

A. *Rule Evolution Experiment*

This experiment was performed to improve car control logic, using GA(Genetic Algorithm). Fitness was calculated in reference to the distance between the car driven on the road and the road centerline and steering frequency. In terms of experiment setup, the number of evolutionary generations was set at 200 and the number of entities in a given generation was set at 20. Following figure shows the number of entities per generation that indicated the highest fitness. Improvement of rule fitness by evolution can be confirmed in this experiment.

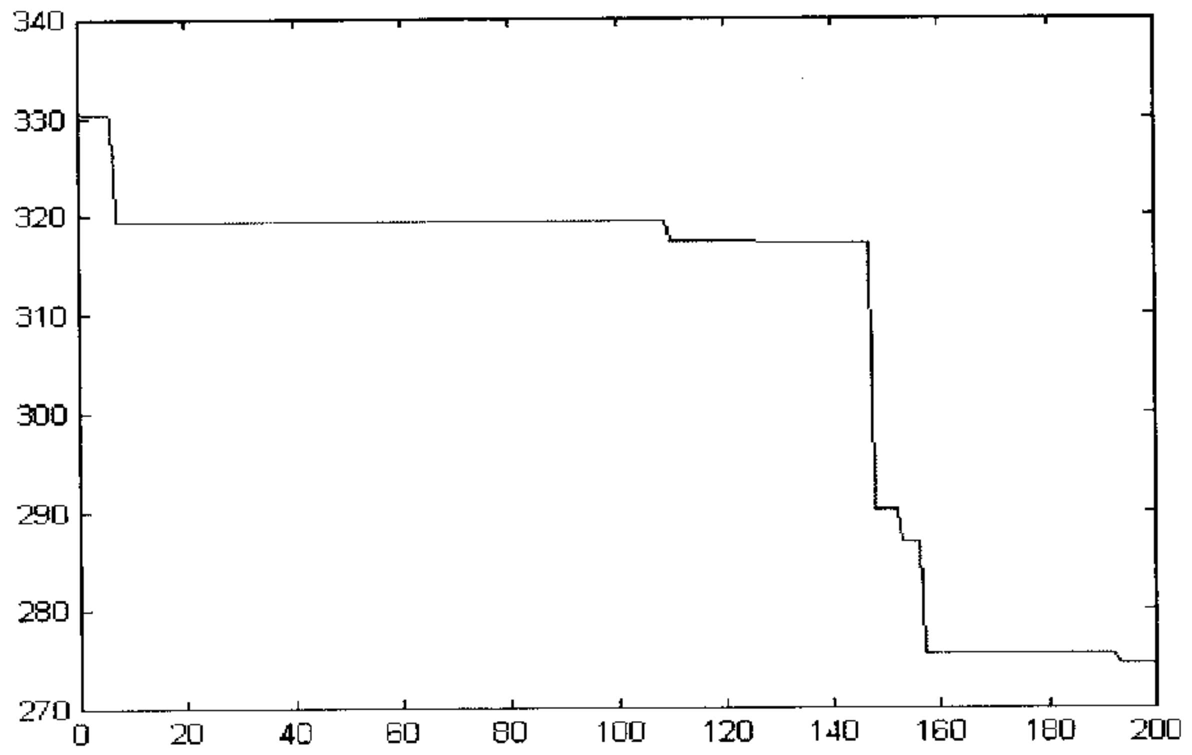


Fig. 10 Fitness Evolution Trend

B. *Driving Experiment*

In this driving experiment, a car controlled by man-defined rules and the other controlled by the rules improved by GA(Genetic Algorithm) were compared.

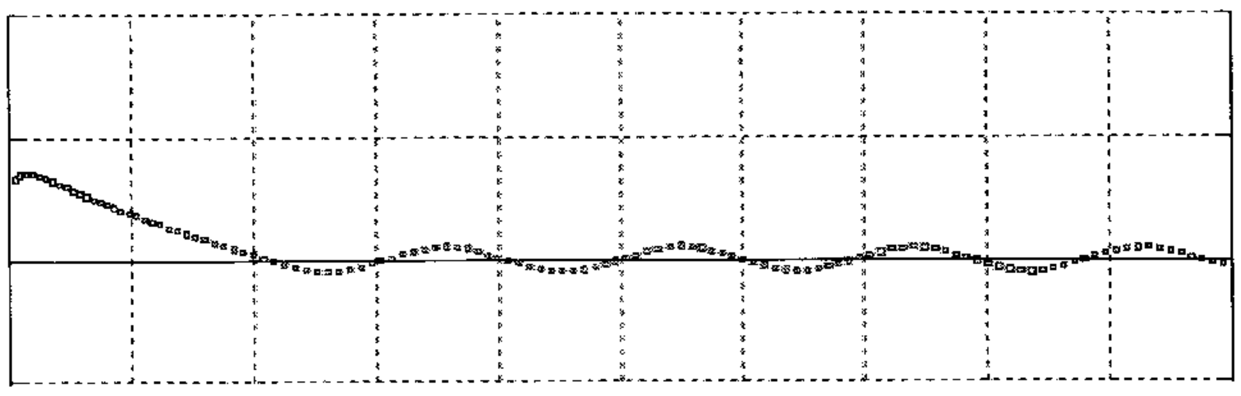


Fig. 11 Movement of Car Controlled by Man-Defined Rules Along Straight Road

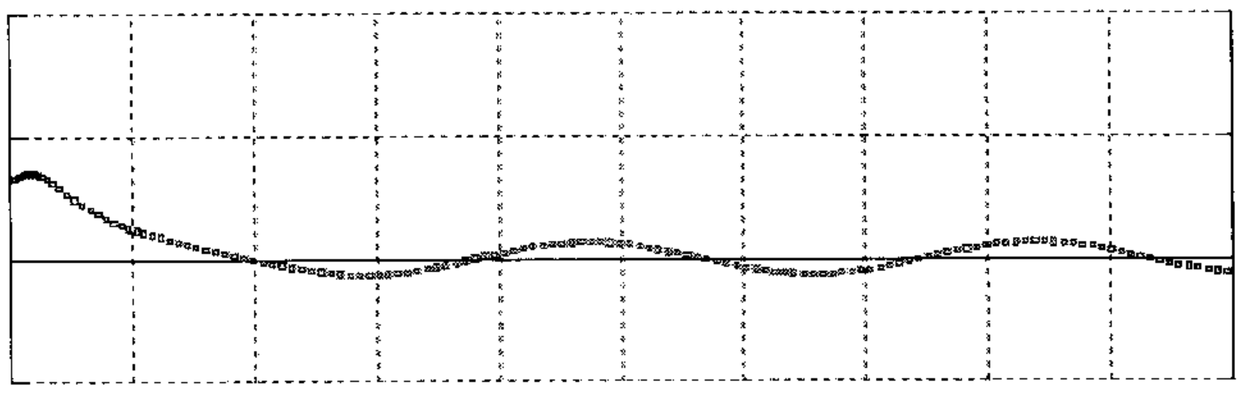


Fig. 12 Movement of Car Controlled by GA-Improved Rules Along Straight Road

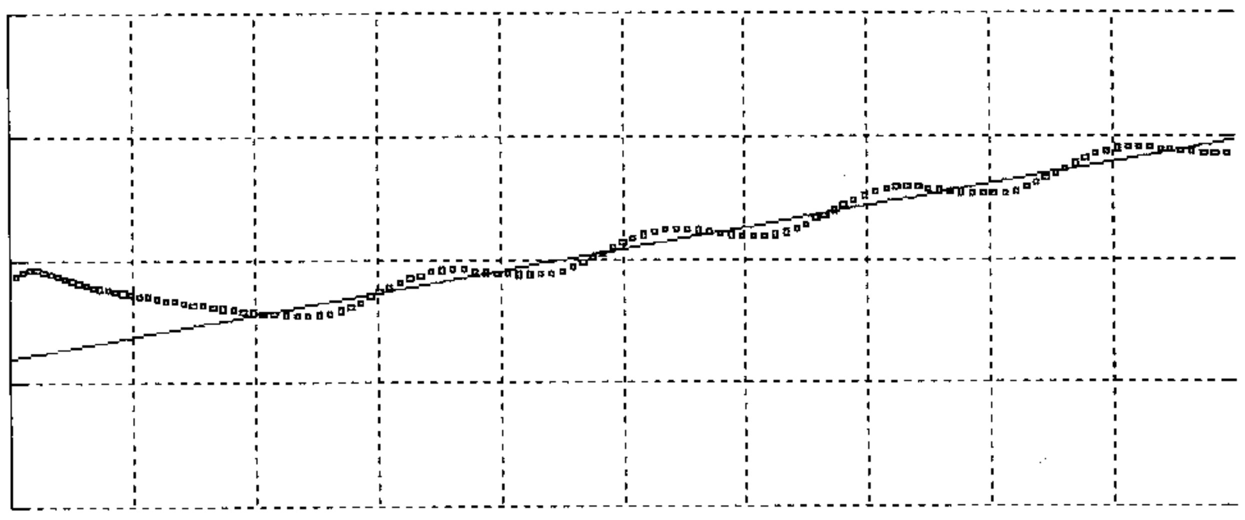


Fig. 13 Movement of Car Controlled by Man-Defined Rules Along Cater-Cornered Road

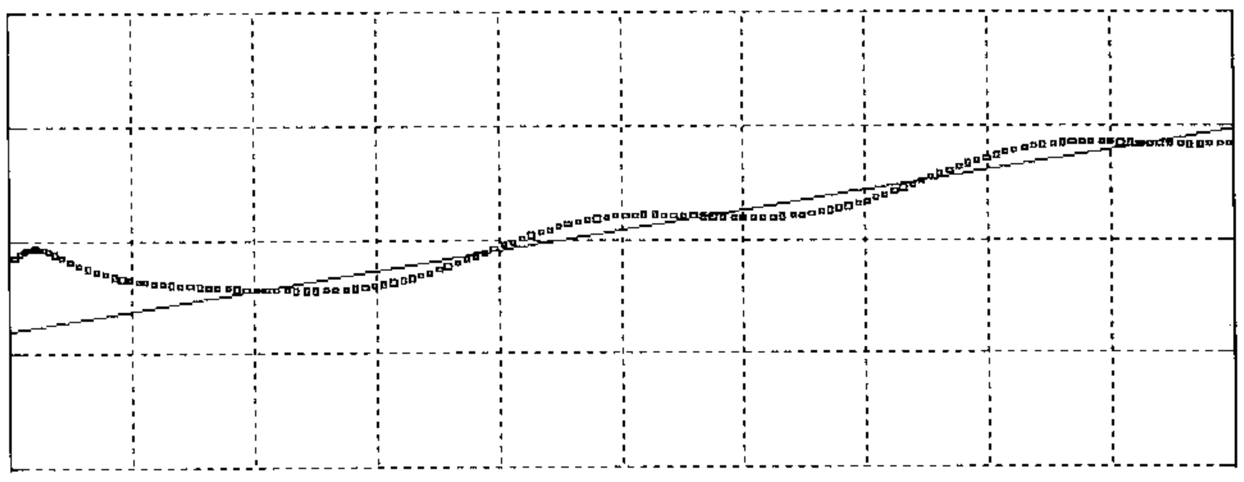


Fig. 14 Movement of Car Controlled by GA-Improved Rules Along Cater-Cornered Road

I applied the road data used in the aforementioned experiments to a simple road alignment mostly consisting of straight sections and a complex road alignment composed of both straight and rotating sections to assess its suitability to diverse road alignments.

The simple road alignment mentioned in the above consists of 40 meter-long straight section followed by short rotating section and cater-cornered section stretched up to 100 meter from the starting point. Then the road goes straight up to its end. The complex road alignment is 1,000 meters long on the X axis with rotating section from 850 meter to 950 meter from the beginning.

The initial car location was defined as (X:0,Y:20, Steering Angle:80), indicating that the car was 20 meter away from the road centerline and the car wheels were positioned at 80 degrees from the road. In other words, the car was off the center of the road and facing a different direction from that of the road. The driving results shown below indicate that the car moves closer toward the road centerline over time.

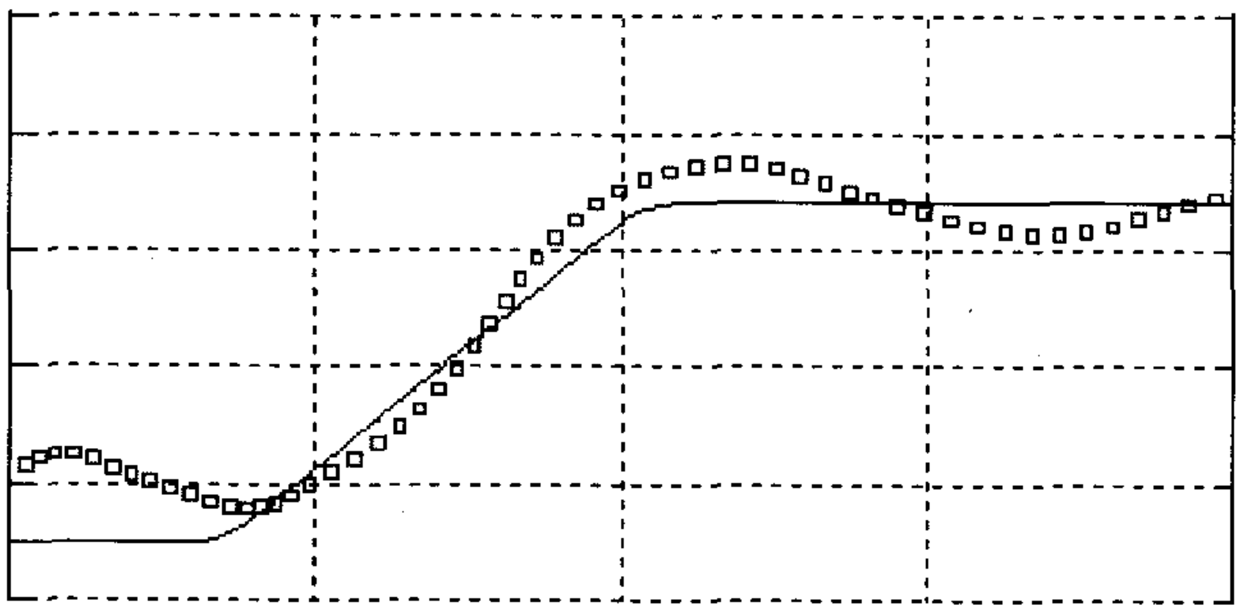


Fig. 15 Car Controlled by Man-Defined Rule along Simple Road Alignment

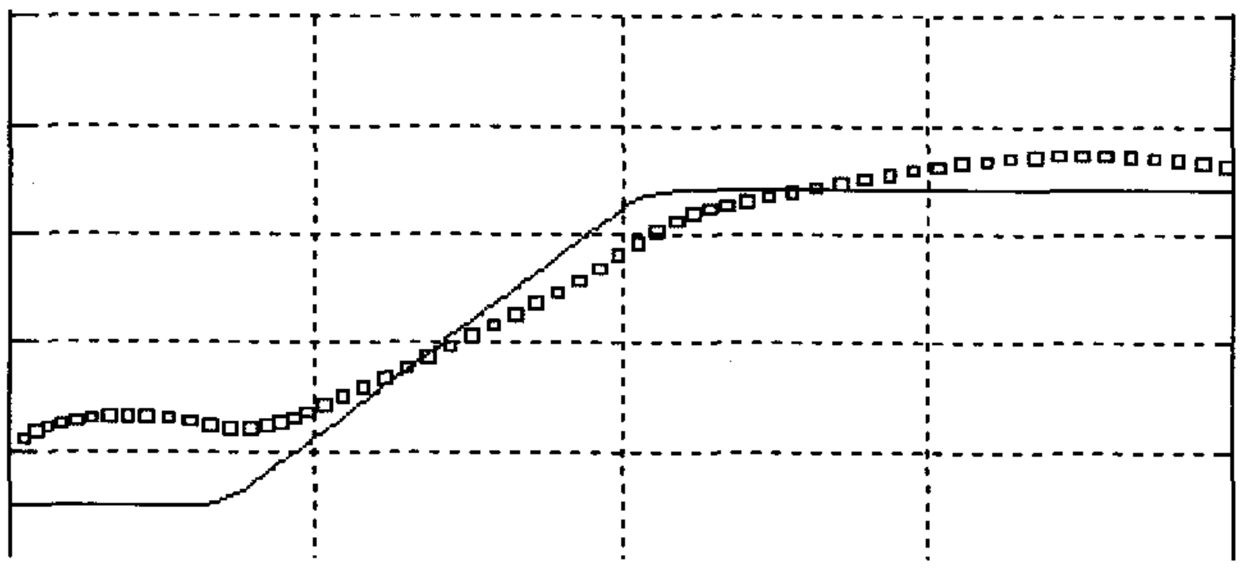


Fig. 16 Car Controlled by GA-Improved Rule along Simple Road Alignment

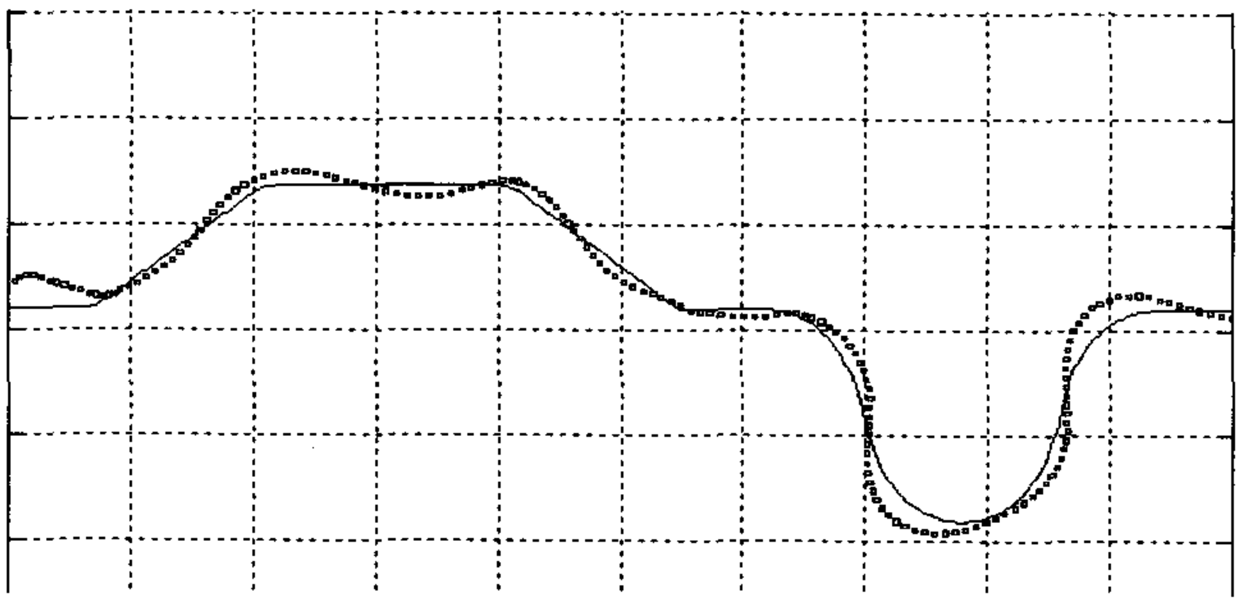


Fig. 17 Car Controlled by Man-Defined Rule along Complex Road Alignment

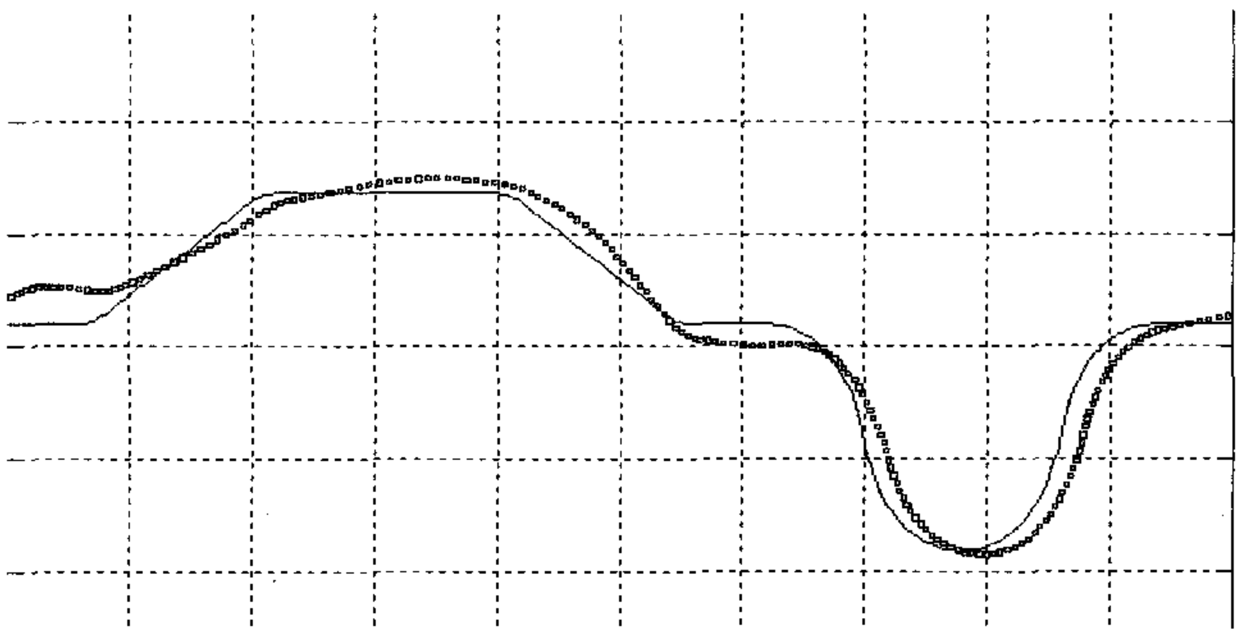


Fig. 18 Car Controlled by GA-Improved Rule along Complex Road Alignment

C. *Driving Experiment Analysis*

Fig. 15 indicates that the car was steered abruptly to move closer to the road center while Fig. 16 shows that the car was steered smoothly by GA-improved rule and the steering frequency was greatly reduced. In addition, the moving distance was greatly reduced in absolute terms in Fig. 16 when compared to Fig. 15. The distance from the road center was a bit longer in Fig. 16, but, the driving line was smoother and car controllability was enhanced as shown in the figures presented in the above.

Fig. 17 shows that the car was steered unnecessarily, yawing even when it was close to straight section. However, Fig. 18 indicates that the car was not steered unnecessarily and rotated smoothly when it approached curved section from straight section. As for the 100 meter-long rotating section placed at the end of the road, it was obvious in Fig. 17 that the car was not driven smoothly while Fig. 18 shows that the car was steered very smoothly.

VII. Conclusion

To control a car, using fuzzy logic controller, fuzzy rules and membership functions should be defined. However, it is not easy to discern necessary rules from unnecessary ones. In addition, it is difficult to find the most optimum membership functions. As a solution to the issues described in the above, genetic algorithm was used herein to define fuzzy rules and membership functions. As illustrated by the results herein, genetic algorithm was able to find fuzzy rules and membership function required for controlling a car in an effective and flexible manner.

REFERENCES

[1] Chuen Chien Lee, "Fuzzy Logic Control Systems: Fuzzy Logic Controller-Part I", IEEE Transactions on Systems, Man, and Cybernetics, vol. 20, No. 2, 1990

[2] Schin-ichi Horikawa et al., "A Fuzzy Controller Using a Neural Network and Its Capability to Learn Control Rules," Proc. of the International Conference on Fuzzy Logic & Neural Networks, pp.103-106, 1990.

[3] Hasegawa T., Horikawa S.-i. Furuhashi T., Uchikawa Y., "On design of adaptive fuzzy controller using fuzzy neural networks and a description of its dynamical behavior", Fuzzy Sets And Systems, Vol. 71, Issue 1, pp.3-23, 1995

[4] F. Hoffmann, et al., Evolutionary Learning of Mobile Robot Behaviours, First Workshop on Frontiers in Evolutionary Algorithms, 1997

[5] Leonid REZNIK, "Evolution of Fuzzy Controller Design", FUZZ IEEE, 1997

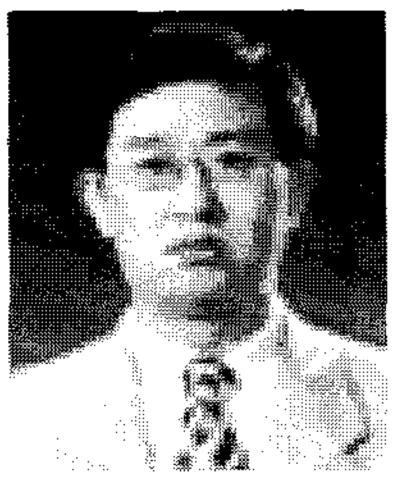
[6] Tomonori HASHIYAMA, Takeshi FURUHASHI, Yoshiki UCHIKAWA, "A Study on Finding Fuzzy Rules for Semi-Active Suspension Controllers with Genetic Algorithm", First Online Workshop on Evolutionary Computation, 1995

[7] Frank Hoffmann, Gerd Pfister, "Evolutionary Design of a Fuzzy Knowledge Base for a Mobile Robot", International Journal of Approximate Reasoning ,vol. 17, no. 4, pp.447-469, (1997)

[8] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Third, Extended Edition, Springer-Verlag, 1995

[9] Potter M. A., De Jong K., "A Cooperative Coevolutionary Approach to Function Optimization", Parallel Problem Solving from Nature-PPSN III, pp.249-257, 1994

[10] Cheol Min, Yoon, "Fuzzy Control Rule Creation Using Genetic Algorithm", Computer Engineering Master's Thesis, Soongsil University, 1998

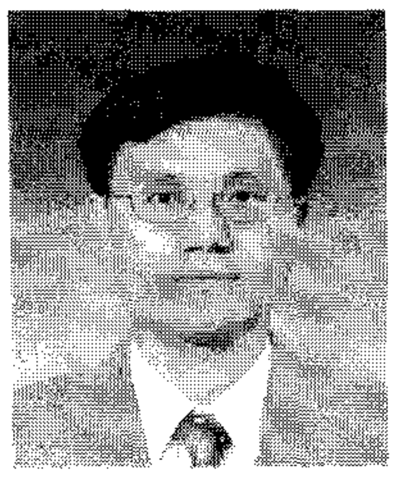


Bong-Gi Kim

He received the B.S., M.S., and Ph.D. degrees in computer science from Soongsil University, Korea in 1987, 1989, and 1999 respectively. Since 1999, he has been with Jinju National University, Korea, where he is currently associate professor in the Division of Computer Science & Engineering. His research interests include multimedia system, home networking and mobile system.

Jin-Kook Song

He received the B.S., M.S., and Ph.D. degrees in computer science from Hongik University, Korea in 1988, 1990 and 1998, respectively. Since 1998, he has been with Jinju National University, Korea, where he is currently associate professor in the division of computer science and engineering. His research interests include programming language, compiler, and reverse engineering.



Chang-Doon Shin

He received his B.S. degree in computer science from Soongsil University in 1989. He received Ph.D. in computer science from Soongsil University in 2004. He is an Associate Professor of Visual Contents at Hallym College. Korea. His primary research interests include Digital Watermarking, Computer Application.