International Journal of Fuzzy Logic and Intelligent Systems, vol. 2, no. 3, December 2002 pp. 242-247

Self-Organizing Fuzzy Controller Using Command Fusion Method and Genetic Algorithm

Na Young-nam*, Choi Wan-gyu**, Lee Sung-joo**

* Dept. of Infomation & Communications, Chosun College of Science & Technology

** Dept. of Computer Science, Chosun University

Abstract

According to increase of the factory-automation(FA) in the field of production, the importance of the autonomous guided vehicle's(AGV) role has also increased. This paper is about an active and effective controller which can flexibly prepare for changeable circumstances. For this study, research about an behavior-based system evolving by itself is also being considered. In this paper, we constructed an active and effective AGV fuzzy controller to be able to carry out self-organization. To construct it, we tuned suboptimally membership function using a genetic algorithm(GA) and improved the control efficiency by self-correction and the generation of control rules.

Key Words : Fuzzy Control, Genetic Algorithm, Fusion Method, Autonomous Guided Vehicle

I. Introduction

Due to the increase of FA's importance in the field of production, the importance of the AGV's role has also increased. The study is about an active and effective controller which can prepare for changing circumstances as they progress. For this study, the research about a self-evolving behavior-based system is also actively being conducted.

In this paper, we proposed an active and effective AGV controller using a fuzzy controller which is able to perform self-organization. For forming the fuzzy control to perform self-organization, we tuned the membership function to the most optimal using a genetic algorithm(GA) and improved the control efficiency by the self-correction and generation of control rules.

The existing AGVs employed a differential and integral calculus(D&I) type controller whose intuitiveness, were flexibility and controlling efficiency was lower[1]. Therefore, it didn't have the ability to actively prepare for the changeable circumstances. However, the self-organizing control(SOC), the fuzzy controller is capable of learning and adapting. It can also generate and correct the control rules. It can then flexibly prepare for changing circumstances and can effectively express the ambiguous and approximate phenomena in the real world by using fuzzy logic. It can make an intuitional controller by using linguistic expressions.

The SOC fuzzy controller proposed in this paper is capable of self-organizing by combining the characteristics of a fuzzy controller and a genetic algorithm. It intuitionally controls the AGV and can easily adapt to circumstances as a human world.

This study was supported(in part) by research funds from Chosun University, 2002

II. Fuzzy Controller

The Fuzzy set is a useful method to describe qualitatively such ambiguous linguistic values as "large" or "small". Decision-making under ambiguous circumstances can't be explained using classical logic, it must be explained by the introduction of fuzzy logic. Fuzzy logic is a representative tool which can artificially make a human-like decision. It can realize an expert system which performs the same functions as human experts, since it describes the experiential knowledge by "if-then" rules and then artificially implementing the mechanism of human decision- making. Since fuzzy logic imitates the ability of human decision- making, it is applied in a variety of areas such as technology, social science, and medical science[2,3].

2.1 The determination of input variables and fuzzy values in the fuzzy control

The fuzzy inference used in this paper can be expressed as a Multiple Input, Single Output(MISO) rule set, which has the rule base as follows[4,5].

R1 : IF X is A(1) and Y is B(1)
THEN Z is C(1), also
R2 : IF X is A(2) and Y is B(2)
THEN Z is C(2), also
.
.
.
Rn : IF X is A(n) and Y is B(n)
THEN Z is C(n).

Here, A(1), A(2),···, A(n) and B(1), B(2),···, B(n) and C(1), C(2),···, C(n) are the fuzzy sets of linguistic terms of input variables X, Y and output variable Y. The above expression is transformed into a fuzzy relation as follows[6,7].

$$R = \text{also}\ (R1,\ R2, \cdots,\ Rn\ )$$

The two variables used in an "if-clause" express the AGV's distance d from the target line and its moving direction θ.

An output variable used in a "then-clause" expresses the AGV's handling direction.

Where x and y represent the present position of AGV, Xp and Yp show the position's point of the most near object-line in AGV. On means an angle of tangent line in the point of Xp and Yp. Here, the fuzzy sets for each input and output variable is as follows.

D = AGV's distance from the target line.
D $\in$ {LF(Left Far), LN(Left Near), MD(Middle), RN(Right Near), RF(Right Far)}

The fuzzy variable LF indicates that the AGV runs far apart from the target line to the right. LN means that it runs near the left side of the target line. MD means that it runs on the target line. RN means that it runs near the right side of the target line.

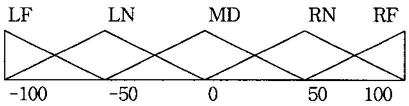


Fig. 2.1 The membership function for distance from target path

V = AGV's navigating direction.
V $\in$ {LN(Large negative), SN(Small Negative), MD(Middle), SP(Small Positive), LP(Large Positive)}

The fuzzy variable LN shows that the AGV is moving to the left side of the target line. MD means that it is moving in the direction of the target.

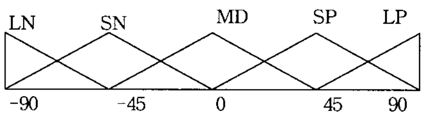


Fig. 2.2 The membership function for the moving direction of AGV

A = AGV's handling angle.
A $\in$ {LN(Large Negative), SN(Small Negative), ZE(Zero Equal), SP(Small Positive), LP(Large Positive) }

The fuzzy variable LN means to turn the AGV's handle far to the left. ZE means to move straight without touching the handle. SP means that the AGV's handle should be turned slightly to the right.

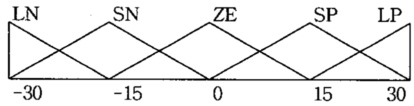


Fig. 2.3 The membership function for the handling angle of AGV

Although each step must be divided into more details for fine control, we divided them into steps as the above figures for the intial steps of our experiment.

Generally, triangles and trapezoids or Gaussian functions are used as the shape of a membership function, but we can use a triangle in this paper. We can use a correlation minimum method for fuzzy inference and a centroid defuzzification method for defuzzification.

In this paper, the AGV is controlled by fuzzy rules which have two input variables. The controller's character and efficiency are determined by the number of input variables or the shapes of their membership functions, so their selection must be made properly. The experienced operator's knowledge and the control expert's knowledge or real data are used for fuzzy partition, since there are no general algorithms and laws for it. Because it is difficult to describe the experiential knowledge completely, a "Cut-and-try" method is required to find efficient rules. In this paper we used the GA for their optimal decision.

2.2 Fuzzy Control Rule

There is "state evaluation" and "object evaluation" in the fuzzy control rule used for fuzzy controller design. The state evaluation is generally introduced in fuzzy control, and it is also used for simulation in this paper. Its general type is as follows.

Ri : If x is Ai, ... , and y is Bi
then z is fi(x, ... , y)

The object evaluation is called the predictive fuzzy control rule, its general form is as follows.

Ri : If(u is Ci → (x is Ai and y is Bi)
then u is Ci)

If this control rule is used, we can predict the next state of plant and control it more smoothly.

These are the methods for extracting the fuzzy control rule:

1) The method introducing the expert's experience and knowledge.
2) The method extracting a operator's function.
3) The method using the controlled process's fuzzy model.
4) The method adding learning ability to the controller.

DIR\DIST	LF	LN	MD	RN	RF
LN	LP	LP	LP	SP	ZE
SN	LP	SP	SP	ZE	SN
MD	LP	SP	ZE	SN	LN
SP	SP	ZE	SN	SN	LN
LP	ZE	SN	LN	LN	LN

Fig. 2.4 control rule

The aim of this study is to organize an active controller which has learning and adapting ability. After giving heuristic control rules, we composed the SOC fuzzy controller by using a genetic algorithm, which can revise the existing control rules or generate new control rules.

2.3 Fuzzy inference and defuzzification

Inference means to extract new facts or relations from the given facts or relations. It can get the new controller's outputs combined using the controller's inputs and control rules. In this paper the Max-Min rule of the various inference methods was used. It is proposed by E.H Mamdani. Its implication and inference are as follows.

$$\mu R(e,ce,co) = Min(\mu E(e), \mu CE(ce), \mu CO(co))$$
$$\mu CO'(co) = Max\text{-}Min(\mu E(e'), \mu CE(ce'), \mu CO(co))$$

e', ce' and co' are fuzzy sets of input and output values of each controller and the inferred output values. μ is the value of member function.

Since the final value gained through fuzzy inference can't be a determinative value for plant control, we can make it a constant to use as a plant input. We call the above course defuzzification.

The following are the most common methods for defuzzification: (1)Simplified center of gravity(COG), (2)Center of gravity method (3)Max criterion method and (4)the Mean of maxima.

In this paper, we employed the COG method which is generally used for defuzzification.

After quantifying the representative set u, the defuzzification value u* is defined as follows.

$$u^* = \mathrm{def}\,uzzifier(B') = \frac{\sum_{j=1}^{n} \mu_{B'}(u_j) \cdot u_j}{\sum_{j=1}^{n} \mu_{B'}(u_j)} \quad (1)$$

III. Genetic Algorithm

A genetic algorithm(GA) is a calculation model which is based on the evolutional phenomena of the real world. It expresses likely solutions to the problem as genetic data structure of a specific form, and gradually transforms them into better solutions. This course is similar to survival of the fittest and prepotency. It is used as a means of search, optimization and machine learning. Its structure is simple, but it is applied to various areas. It is an effective method for finding an approximate value for optimization. These days, it is actively being combined with the various theories. The genetic algorithm which imitates the evolutional course of an organism basically keeps the genetic types within a group for problem solving. It generates the new genetic types through the operator like crossover or mutation, it selects the genetic types which have a high degree of benefit for the group, and forms a new group composed of the selected new genetic types. And it solves the problem using them. The basic genetic algorithm is as follows.

```
[step 1] : initialize(population)
           evaluation(population)
[step 2] : if termination-condition = O.K.
           then end
[step 3] : reproduction(population)
           alter(population)
[step 4] : goto step 2
```

Fig. 3.1 The basic genetic algorithm

[Step 1] generates each initial value randomly in the range of the fittest variables, which is the potential solution set population composed of a single chromosome. It forms the initial potential solution set, and reproduces objects which have high fitness in the next generation. [Step 3] alters the individuals of the reproducted group and repeats the global search by using mutation and crossover. To alter(population) means to mutate and to crossover the reproduced potential solution sets to ensure high probability. [Step 2] is closed after a specific generation, or it is closed when there is no improvement after the regulated generation has passed.

3.1 The genetic expression types

In this paper, we expressed the parameters of the fuzzy system used to control. The AGV as genetic types, generated new genetic codes by the mutation and the crossover of the parameters having the high fitness, applied them to the fuzzy system, and obtained the fittest parameters.

The structure of the genetic types are as follows.

2 input-vars	1 output-var	rules
INPUT	OUTPUT	RULES

chromosome

Fig. 3.2 The structure of genetic

That is, two fuzzy input variables, a output variable and the control rules are encoded. The encode of the fuzzy input and output variables needs 36 bits, since each variable of input and output needs 12 bits to determine the membership function as in Fig. 3.3.

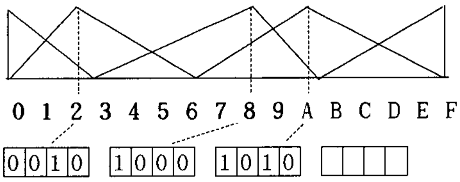


Fig. 3.3 The coding method of the membership function.

3.2 The coding method of control rules

To code the control rules as chromosomes, we substitute a control rule with integers, and arrange it in the previously fixed order. To code the fuzzy controller as the genetic algorithm, after coding each input and control variable according to the coding methods of the membership functions and the control rules, we should arrange them in a chromosome and arrange the coded control rules next to them.

Fig. 3.4 coded the application of genetic algorithm for the diversion of binary digit(from 000 to 100) for LP, SP, ZE, SN, and LN in the control rules of Fig. 2.4.

DIR/DIST	LF	LN	MD	RN	RF
LN	000	000	000	001	010
SN	000	001	001	010	011
MD	000	001	010	011	100
SP	001	010	011	011	100
LP	010	011	100	100	100

↓

000 000 000 001 010 000 001 ··· 100 100

Fig. 3.4 The coding method of control rules

3.3 The genetic operator

1. Reproduction : As a method to make a fit individual, produce more sons in the next generation, the dominance selection method is used.
2. Crossover : In the reproduced parental individuals, select a part of string arbitrarily and cross them over.
3. Mutation : Randomly generate a value between 0 and 1 for each individual gene in each set. If the generated value is less than that of the mutation actor, alter the value of the gene.

   Genetic algorithm's application environment used simulation of this paper include the following.

- pop_size=20
- Fitness's function:

$$Fitness_i = \frac{1}{1 + \sum_{i=0}^{216} |error_i|} \quad (2)$$

- Error's value: errori = Ei - Fi
- reproduction method: The method of Roulette

$$\rho_i = \frac{Fitness}{\sum_{j=0}^{100} Fitness_j} \quad (3)$$

(ρ: *the probability of reproduction*)

- maximum fitness: 0.99
- pc=0.25, pm=0.01

3.4 The mutation of the control rules

The gene encoded control rules cause mutations as a constant probability, similar to the mutation operation of the general genetic algorithm. The control rules of the conclusion part can be expressed as any of a number of linguistic terms. Therefore, they have the new control rule which has one of the remain values, if mutation occurs.

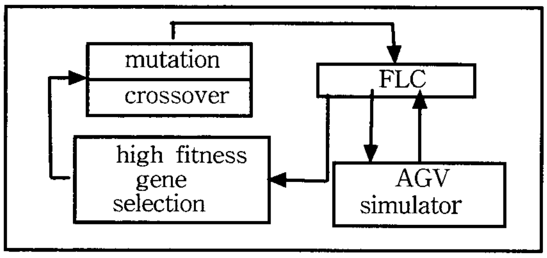


Fig. 3.5 The tuning structure of the fuzzy system by GA

Fig. 3.5 show the structure of tuning in the parameter of the fuzzy system by a genetic algorithm.

IV. A simulation and the results

4.1 Simulation

In this simulation, the proposed self-organizing fuzzy controller optimally learned the fuzzy membership functions and rules, and performed the fuzzy control. The speed was assumed to be constant.

Fig. 4.1 shows the input and output state variables of the AGV used in the simulation. The distance d from the target line and the vehicle's navigating direction θ are the conditional variables and the handle's turning angle θ is the conclusive variable. Variable d expresses the distance between the AGV and the target path, the sign of d is determined by the AGV's position.

If the AGV is on the left side of the target path, sign of d is "-". If the AGV is on the right side, sign of d is "+".

The length of d is transformed into a fuzzy variable and inputted as a linguistic value.

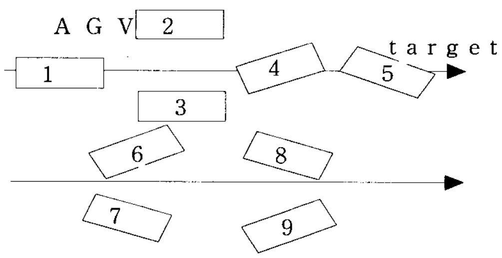


Fig. 4.1 The target path and the location of AGV

Fig. 4.1 illustrate 9's rule for a combination d and θ. In order to evaluate the navigating efficiency, error e, $\triangle$d and

△e are used as inputs of function evaluation. The AGV's navigating efficiency is measured using this function.

θ expresses the AGV's navigating direction. We can calculate the relation between the AGV and the target path by comparing θ with θp. It is possible to make a fuzzy rule like Fig. 4.1 by using this relation. Fig. 4.1 expresses the location relationship between the AGV and the target path. The control rules are illustrated in Fig. 2.4. The simulation was performed on the local path as in Fig. 2.4 using the discussed genetic algorithm and fuzzy inference.

4.2 AGV's path planning and navigation

The following algorithm expresses the AGV's global navigation algorithm.

[step 1] : Making a global map of workspace
[step 2] : Global path planning
[step 3] : Local navigation of AGV
(1) IF current position = goal point THEN navigation terminates
(2) IF an obstacle exists THEN obstacle avoidance, local path planning, go straight to sub-goal point, path recovery.
(3) go straight to goal point
[step 4] : go to step 3

The AGV's pilot module makes the AGV navigate toward the target on the path generated by the global planning. In this paper, we used the path planning seen in Fig. 4.2 for the simulation.

It is assumed that the information of workspace is already known. We evaluated the AGV controller's performance by using the above path planning. The lateral position error, as a function to evaluate the navigating performance, expresses its displacement from the target path. The absolute value of the displaced distance is defined as e.

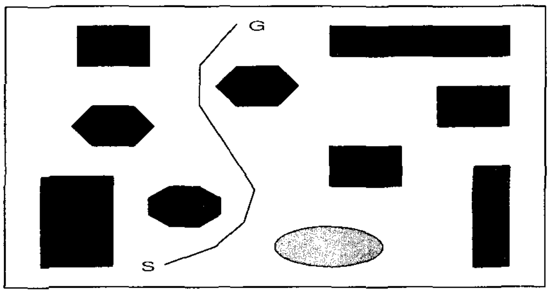


Fig. 4.2 simulation 1

The total lateral position error at a specific distance is defined as its evaluating function. The lower the value of the evaluation function is, the closer the AGV navigates to the target path.
Fig. 4.3 show the result of simulation in the different environment for an experiment of the various environment.

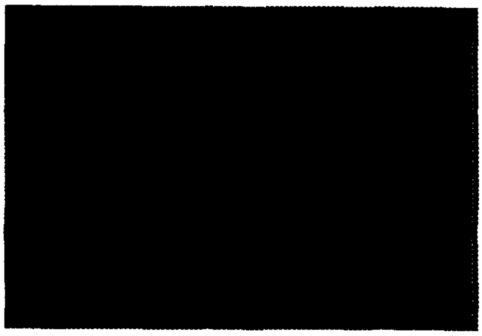


Fig. 4.3 Simulation 2

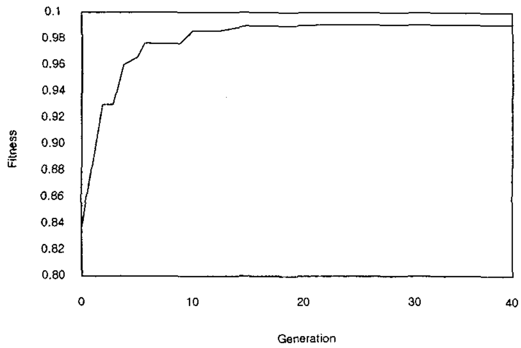


Fig. 4.4 Fitness history

Fig. 4.4 represent the change in graph of fitness. Fitness history started 0.84 for the initial fitness, fuzzy system regulated by genetic algorithm represented the fitness of over 0.99, finally. Fuzzy system regulated the parameter's rule by genetic algorithm shared more improvement than fuzzy system of existing.

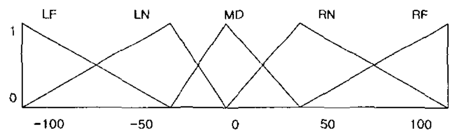


Fig. 4.5 An example of the adjusted membership function

Fig. 4.5 illustrates to us the member-function, or the fittest of the functions generated by the proposed method. From the comparison, the member-function's shape and control rules generated by heuristic method with the membership function's shape and control rules initially generated by the genetic algorithm, we can see that the performance of a non-linear control is higher than that of linear control. It improved the ability to operate the handle as follows : if the error between the AGV and the target path is wide, move a large amount and if the error is small, do it slightly

The axis of x in the three graph represent the change of time, y's axis signed the magnitude of error e, accummulation of e, the change of Δe, These used the traveling performance and the evalution of stability. Fig. 4.6.

Fig. 4.6 (a) graph of error function
(b) cumulative graph of error function
(c) graph of Δe function

V. conclusion

In this paper, we proposed the navigating controller which can be applied to the AGV's navigation by forming a self-organizing fuzzy control system for the AGV's efficient control. Since the control algorithms of the existing AGVs only depended on the knowledge of the expert, there were limitations in the optimization of parameters and the AGV's efficiency. Also it was very difficult to understand instinctively because of too many rules and the complicated calculations. Therefore we composed a controller which is instinctive and easy to understand. This controller is the fuzzy system which uses linguistic variables. For the composing controller to be able to actively prepare for the changeable circumstances, we optimized the membership-function by using the SOC fuzzy controller. We implied employing the genetic algorithm and ensured the fittest possible navigation through self-correction and the generation of control rules.

We can obtain a highly efficient fuzzy control in the case of off-line by applying fuzzy control rules using a genetic algorithm, but it cannot be applied to real time. A model which can be applied to real time is needed to actively prepare for changeable circumstances. We will continue to study the model, which can learn at real time and will be useful in avoiding arbitrary obstacles generated during the AGV's navigation.

References

[1] In-Yong Lee, Han-Ho Tack, "Control of Flexible Joint Robot Using Direct Adaptive Neural Networks Controller", international journal of fuzzy logic and intelligent systems", pp.29-34, 2001.

[2] Domg Hwa Kim, Jin ILL Park, "Auto-Tuning of Reference Model Based PID Controller Using Immune Algorithm", journal of fuzzy logic and intelligent systems", vol.12 n.13 pp.245-254, 2002.

[3] Byung-Jae Choi, "Design of Single-input Direct Adaptive Fuzzy Logic Controller Based on Stable Error Dynamics", international journal of fuzzy logic and intelligent systems", pp.44-49, 2001.

[4] B. McCane and T. Caelli, "Multi-scale adaptive segmentation using edge and region based attributes", Proc. KES '97, University of South Australia, pp. 72-80, 1997.

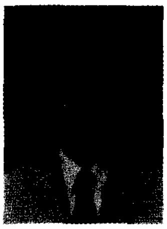
[5] Atsushi Konno, Masaru Uchiyama, "Modeling of a flexible Manipulator Dynamics Based upon Molzer's Model, "IROS '96, Japan, pp. 223-229, 1996.

[6] K. Le, "A proposed model of human consciousness system with applications if pattern recognition", Proc. KES '97, university of south Australia, pp. 159-167, 1997.

[7] Karem Zita Haigh, Manuela M. veloso, "Interleaving Planning and Robot Execution for Asynchronous user Requests", IROS '96, Japan, pp. 148-155, 1996.

Young-Nam Na
1991 : Dept. of Computer sciences, Chosun University(B.S)
1993 : Dept. of Computer sciences, Chosun University(M.S)
1998 : Dept. of Computer sciences, Chosun University(Ph.D)
1988~now : Assistant Professor in the Dept. of Information & Communications, Chosun College of Science & Technology

Research Interests : Fuzzy system, Genetic Algorithm, Adaptive Control System, AMR, CAI



Wan-Gyu Choi
1994 : Dept. of Religious Studies, Seoul National University(M.S)
1994 : Dept. of Computer sciences, Chosun University(M.S)
1999 : Dept. of Computer sciences, Chosun University(Ph.D)

Research Interests : Rough set, Software engineering, Programming Language, Object-oriented software.

Sung-Joo Lee
1970 : Dept. of Physics Sciences, Hannam University(B.S)
1992 : Dept. of Computer sciences, Kwang--woon University(M.S)
1998 : Dept. of Computer sciences, Catholic University of Daegu(Ph.D)
1988~1990 : Chief, Computer Center, Chosun University
1995~1997 : Deant, Information Science, Chosun University
1981~now : Professor in the Dept. of Computer Engineering, Chosun University

Research Interests : Software engineering, Programming Language, Object-oriented software, Rough set.