A Self-Organized Fuzzy Controller for Wheeled Mobile Robot Using an Evolutionary Algorithm

Sung Hoe Kim, Chongkug Park, *Member, IEEE*, and Fumio Harashima, *Fellow, IEEE*

***Abstract*—Designing the controller of a wheeled mobile robot is not as easy as might be supposed, on account of nonholonomic constraints. To overcome such difficulties and gain more accurate position and velocity control, a self-organized fuzzy controller is proposed. To find solutions of optimal fuzzy input and output membership functions and to determine a rule base, an evolutionary process is proposed. The procedure that derives this solution is composed of three steps, each step having its own unique evolutionary process. The elements of an output term set are increased first, and then the rule base is varied according to increase of the elements. The varied fuzzy system competes with a system that has no element increase. If the varied fuzzy system loses in competition, then the system naturally disappears. On the other hand, if the varied system survives, the fitness with each increased element of the output term set and rule base is tested and unnecessary parts are removed. After having finished regulation of output term set and rule base, searching for input membership functions is processed with constraints to reduce the unsuitability of the system. The searching constraints do not produce a 0 membership value for any input. After completing the search for the input membership function, fine tuning of output membership functions is processed.**

***Index Terms*—Fine tuning, self-organized fuzzy controller, system competition.**

I. INTRODUCTION

A FUZZY controller is generally designed in the light of experience and expert knowledge [1]. However, such knowledge is not always easy to acquire. Moreover, even where the system is designed by skillful and well experienced experts, whether it is well enough designed to be optimized will be uncertain. To improve this situation, the fuzzy system can be combined with a neural network or an evolutionary process [2]–[4]. Tan and Hu have used a genetic algorithm to derive a systematic design of a fuzzy controller [5]. Hsu has used a multi-operator genetic algorithm to gain optimization of a fuzzy control rule. Castro has shown a method for getting a proper upper bound of rule number for fuzzy controllers [6], [7]. Hsu and Tsai designed a fuzzy-proportional plus integral (PI) controller that can automatically control membership function with a genetic algorithm [8]. However, these algorithms had some problems, in that the algorithms cannot find both optimized rule base and membership function.

Manuscript received January 19, 2000; revised October 19, 2000. Abstract published on the Internet December 18, 2000.

S. H. Kim and C. Park are with the School of Electronics and Information, Kyung Hee University, Seoul 449-701, Korea (e-mail: ckpark@nms.kyunghee.ac.kr).

F. Harashima is with Tokyo Metropolitan Institute of Technology, Tokyo 191-0065, Japan (e-mail: f.harashima@ieee.org).

Publisher Item Identifier S 0278-0046(01)02646-6.

To solve such problems, we propose a new self-organized fuzzy algorithm that is designed through an evolutionary process. Through this evolution process, the parameters in each element of the output term set are updated continuously. After finishing this updating procedure, regulation for the output term set and rule base is processed. Then, similarities between the existing elements of the output term set and elements of the updated output term set are calculated. According to the similarity degree produced by such calculation, new elements are added to the existing output term set. Adding new elements induces variation of the rule base. The fitness of the system with varied rule base and added elements is evaluated. If an acceptable result is not obtained by this evaluation, then the added elements are removed. After the process of removing of surplus elements is completed, a population is formed under the regulated output term set. This process is repeated again and again. After regulation of the output term set and rule base, searching for optimal input membership functions is processed under some constraints. If any output fails to exist for any input, then the system might fall into an uncontrollable state. To avoid such a problem, input membership functions that are bigger than 0 of membership value for all inputs are searched. After the searching of input membership functions is completed, fine tuning for output membership functions is processed. The proposed algorithm has some constraints on application due to the system's characteristics. However, because optimization for rule base and membership function are performed by self-learning, it can be applied to various fields in which conventional fuzzy algorithms and fuzzy-neural networks are implemented. In this paper, we performed computer simulations for a two-degrees–of–freedom (2DOF) wheeled mobile robot (WMR) to verify the feasibility of the proposed algorithm.

II. REGULATION OF OUTPUT TERM SET AND RULE BASE BY EVOLUTIONARY PROCESS

A fuzzy controller is generally composed of fuzzification, rule base, inference engine, and defuzzification. In fuzzification, numerical inputs are transferred as linguistic fuzzy values. A control procedure that is processed in the rule base is depicted as a linguistic description. In the inference engine, a linguistic control value occurs by max–min operation. In defuzzification, linguistic inference values are transformed as numerical outputs.

Fuzzy set F is described as $F = \{u, \mu_F(u) | u \in U\}$ by membership function $\mu_F(u)$. U is the universe of discourse. Language variable x in the fuzzy controller is characterized by term set $T(x) = \{T_x^1, T_x^2, T_x^3, \ldots, T_x^k\}$ and membership function

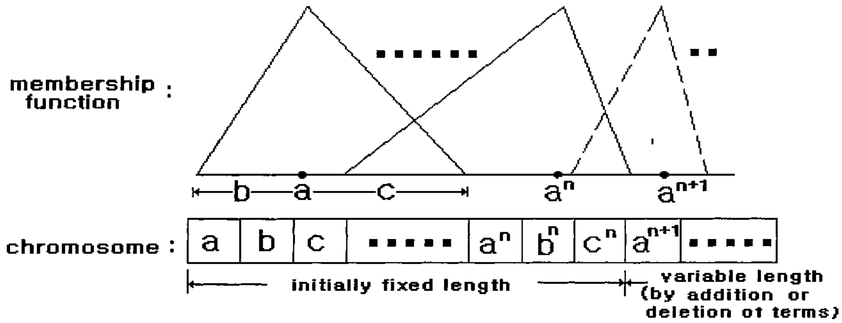

Fig. 1. Coding in chromosome for membership function parameter.

$\mu(x) = \{\mu_x^1, \mu_x^2, \mu_x^3, \ldots, \mu_x^k\}$. The state vector X is defined by a membership function and term set as follows:

$$X = \left\{ \left(x_i, U_i, \{T_{x_i}^1, T_{x_i}^2, T_{x_i}^3, \ldots, T_{x_i}^{k_i}\} \right.\right. \\ \left.\left. \{\mu_{x_i}^1, \mu_{x_i}^2, \mu_{x_i}^3, \ldots, \mu_{x_i}^{k_i}\}\right) |_{i=1, \ldots, n} \right\}.$$

Each input and output membership function is described as

$$\mu^j(x) = \begin{cases} 1 + \dfrac{x - a_j}{a_j - b_j}, & b_j \le x \le a_j \\ 1 + \dfrac{a_j - x}{c_j - a_j}, & a_j \le x \le c_j \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The elements in the output term set are updated by evolutionary process. As shown in Fig. 1, parameters in the output membership function are coded in the chromosome.

The population P_{org} for the evolution process is composed of chromosomes as shown in Fig. 2. In order to get an optimal output term set, an appropriate initial output term set is required. To have such an appropriate initial output term set, an evolution process that involves crossover, mutation, and reproduction is processed. While these procedures are carried out, a large mutation rate is chosen in order to adjust the consequent clause in the initial rule base. After arriving at reference generation, chromosome $A(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$ that has the highest fitness value is selected from the population. Then, new population P_{first} is produced from chromosome $A(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$. P_{first} is composed of chromosome $B(a_{i\_\text{start}}, b_{i\_\text{start}}, c_{i\_\text{start}})$ that is produced by

$$B_{\text{start}} = A_{\text{final}} + \upsilon(t) \times r \tag{2}$$

where

- $\upsilon(t)$ maximum variation value;
- r random value from 0 to 1;
- t number of repeated initializations at population based on searched optimal chromosome after reaching the final generation.

Then, population P_{first} should have a chromosome that is produced when $r = 0$. When a new chromosome is produced, a chosen part in the selected optimal chromosome is applied to produce a new chromosome.

After this new population is completely composed, the evolution process should be restarted. During the evolution process, multi-partial-mapped crossover (MPMC) is used [10]. When the evolution process arrives at the final generation, chromosome $B(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$ that has the highest fitness is obtained. Through comparison of chromosome $A(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$ with chromosome $B(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$, which are all fuzzy membership functions, similarity measure $E(A, B)$ is obtained as [9]

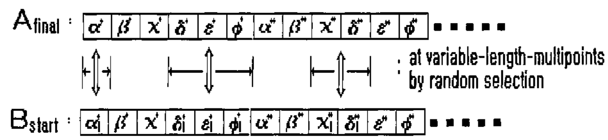

Fig. 2. New chromosome production from optimal chromosome.

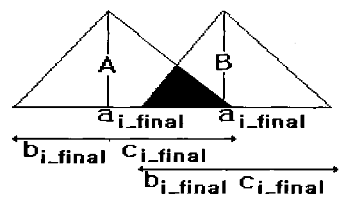

Fig. 3. Similarity measure between two membership functions.

$$E(A, B) = \frac{C(A_{\text{final}} \cap B_{\text{final}})}{C(A_{\text{final}} \cup B_{\text{final}})} = \frac{C(A_{\text{final}} \cap B_{\text{final}})}{C(A_{\text{final}}) + C(B_{\text{final}}) - C(A_{\text{final}} \cap B_{\text{final}})}. \tag{3}$$

$C(X)$ is the cardinality of fuzzy set X.

The similarity measure between two membership functions is shown in Fig. 3.

Adjustment of the rule base based on the similarity measure is processed as follows [9]:

$$degree(i) = \max E[M_A(a_{j\_\text{final}}, b_{j\_\text{final}}, c_{j\_\text{final}}) \\ M_B(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})], \qquad 1 \le j \le k$$

IF $degree(i) < \alpha(t)$
THEN
Insert $M_B(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$ into the term set $A(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$

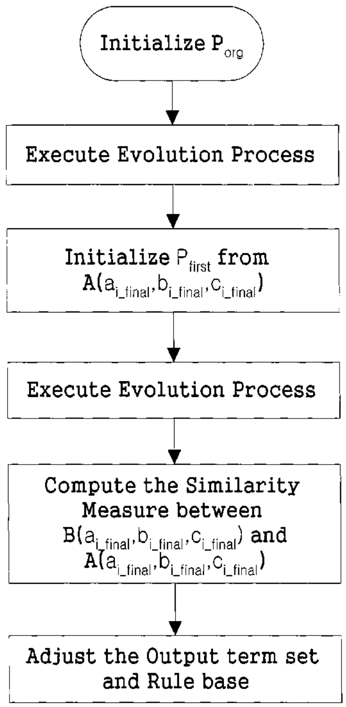

Fig. 4. Adjustment of rule base and output term set.

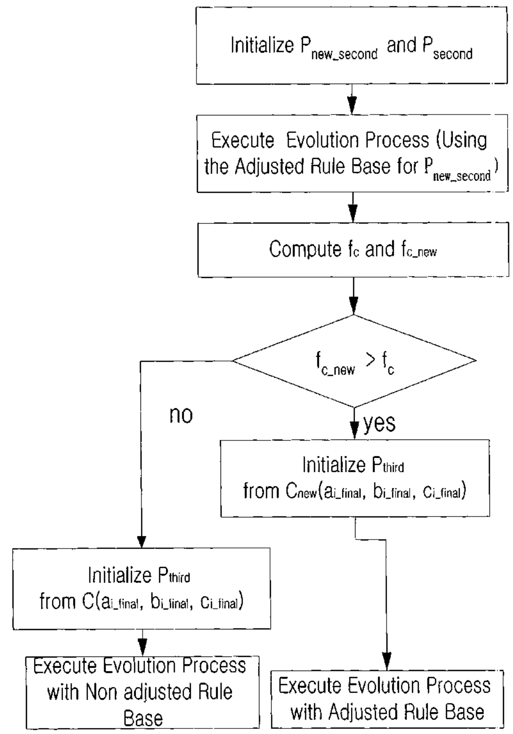

Fig. 5. Flowchart of competition.

IF $q_i \geq \beta$
  THEN
    change the consequence of ith rule from $M_A(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$ to $M_B(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$

where
- $\alpha(t)$ similarity criterion;
- k size of the fuzzy portion of the output linguistic variable;
- q_i firing strength from ith rule;
- β firing strength threshold.

The procedure for adjustment of the rule base and output term set is shown in Fig. 4.

III. System Design Through Competition

The output term set that has increased elements through the similarity measure is described as $C(a_{i\_\text{new}}, b_{i\_\text{new}}, c_{i\_\text{new}})$. New population $P_{\text{new\_second}}$ is initialized as $C(a_{i\_\text{new\_start}}, b_{i\_\text{new\_start}}, c_{i\_\text{new\_start}})$ that is produced by

$$C_{\text{new\_start}} = C_{\text{new}} + \upsilon_{\text{new}}(s) \times r \tag{4}$$

where s is the number of increased elements in output term set. Another population P_{second} is also initialized as $C(a_{i\_\text{start}}, b_{i\_\text{start}}, c_{i\_\text{start}})$ that is produced through

$$C_{\text{start}} = B_{\text{final}} + \upsilon \times r. \tag{5}$$

Only selected elements in the chromosome that has the highest fitness value are related with the production of new chromosomes. The evolution process is performed individually for each population. The adjusted rule base is applied to $P_{\text{new\_second}}$ and the nonadjusted rule base is applied to P_{second}. Such procedure is shown in Fig. 5. Having completed the evolution process, chromosomes $C(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$ and $C_{\text{new}}(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$ that have the highest fitness value are selected from populations P_{second} and $P_{\text{new\_second}}$. Then, fitness f_c for $C(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$ and f_{c_new} for $C_{\text{new}}(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$ are compared together. If $f_c > f_{c\_\text{new}}$, then new population P_{third} is produced from $C(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$ and the nonadjusted rule is applied to the evolution process. If $f_c < f_{c\_\text{new}}$, then P_{third} is produced from $C_{\text{new}}(a_{i\_\text{final}}, b_{i\_\text{final}}, c_{i\_\text{final}})$ and the adjusted rule base is applied to the following evolution process. Such an adjustment for an output term set and a rule base is performed again and again. The overall flowchart that contains the algorithm is shown in Fig. 6.

IV. Searching for Input Membership Functions with Constraints and Fine Tuning of Output Membership Functions

Searching for input membership functions is performed with same method as was done for the searching of output membership functions. Population P_{input} is produced from initial input membership functions with the same method as previously. The evolution process is performed in population P_{input}. While the evolution process is performed, a constraint is applied. When a fuzzy system has input, there must be a matched output. If output for input does not exist, then an uncontrollable situation may occur. To avoid such a problem, the membership value for all input should be larger than 0. Having completed the search for input membership functions with a constraint, fine tuning for output membership function begins.

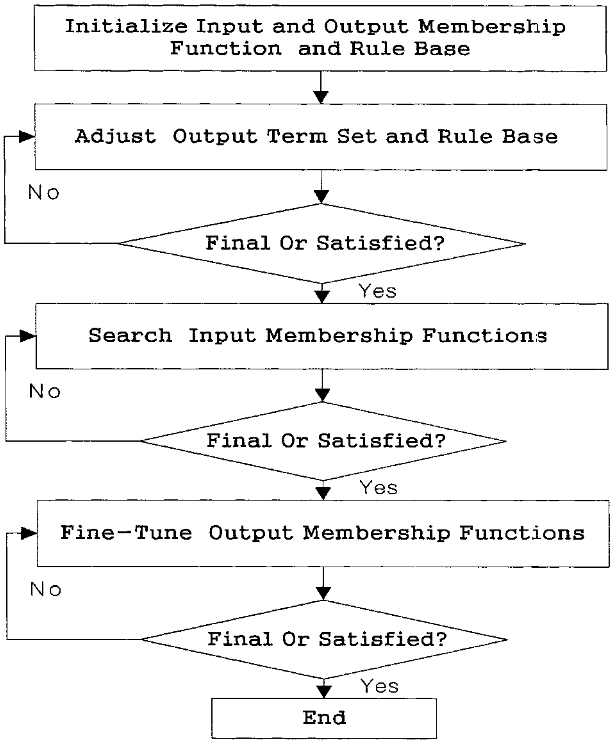

Fig. 6. Overall algorithm for self-organization.

As shown in the previous section, the aim of adjustment is to increase the elements of the output term set and to change the rule base. However, the aim of this section is to fine tune output membership functions. To perform the fine tuning, population P_{output} is initialized by (6) and composed of x^{t+1}. In (6), x^t is the optimal chromosome for membership functions of the output linguistic variable that is obtained by the algorithm shown in the previous section

$$x^{t+1} = x^t + N(0, \sigma(t)) \tag{6}$$

where $\sigma(t)$ is the standard deviation and $N(0, \sigma(t))$ is a random Gaussian number.

Chromosome A_{output} that has the highest fitness value is selected from P_{output}. By (6), P_{output1} is produced from A_{output}. Chromosome A_{output1} that has the highest fitness is selected from P_{output1}. This iterative process performs fine tuning of output membership functions. When a new population is produced with the same method as before, a chosen part in the selected optimal chromosome is applied to produce a new chromosome.

V. Computer Simulation and Results

Computer simulation of the proposed algorithm is performed for a 2DOF WMR. In Fig. 7, a coordinate assignment for a robot is shown. In this figure, $P = [x, y, \theta]$ means each real robot position and rotation angle in robot reference coordinate X, Y [11].

Parameters of the WMR that are used in the computer simulation are shown in Table I.

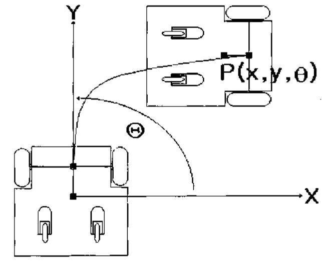

Fig. 7. Coordinate assignment of position and orientation of WMR.

TABLE I
PARAMETERS OF WHEELED MOBILE ROBOT

Values	Dimension	Contents
0.3048	M	Height of body
0.3048	M	Height of load
0.2667	M	Width of body/2
0.2667	M	Length of body/2
0.2667	M	y-displacement of caster
0.1524	M	z-displacement of caster
0.2667	M	x-displacement of caster
0.1524	M	z-displacement of caster
0.0254	M	Length of steering axle
0.1127	M	Radius of driving wheel
0.0381	M	Radius of driving caster
90.72	Kg	Weight(mass) of body

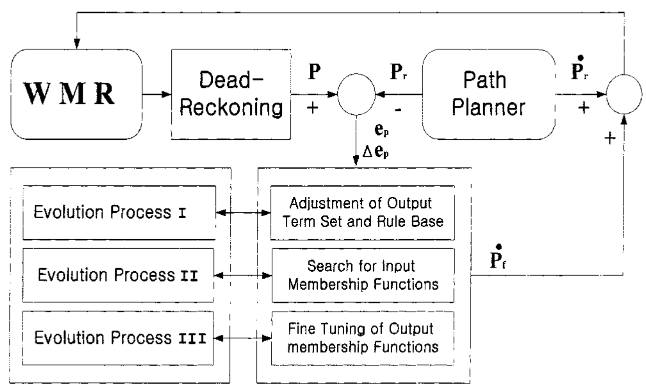

Fig. 8. Overall block diagram of proposed system.

The overall block diagram of the proposed system is shown in Fig. 8.

Input variables of fuzzy controller are an error $e_p = P_r(x_r, y_r, \theta_r) - P(x, y, \theta)$ and an error variation $\Delta e_p = e_p(n+1) - e_p(n)$. Output variables are velocities for X, Y, Θ. The number of elements of the initial term set for input variable and output variable is five each. Each membership function parameter is individually and randomly selected. Population size and length of chromosome are 30 and 90, respectively. Crossover probability P_c and mutation probability P_m are variable and initialized as 0.5 and 0.3. The mutation rate, crossover rate, and population size were chosen by the trial-and-error method. The chromosome was initialized by random value in a limited range. The length of chromosome is extended as much as the number of increased terms through learning from competition. The fitness function $F(t)$ that is used in evolution process is shown in (7). We set up the time variable weighting factor as a monotonic increasing function.

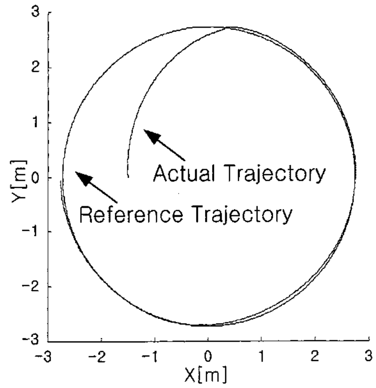

Fig. 9. Real trajectory of WMR for round-shaped reference trajectory.

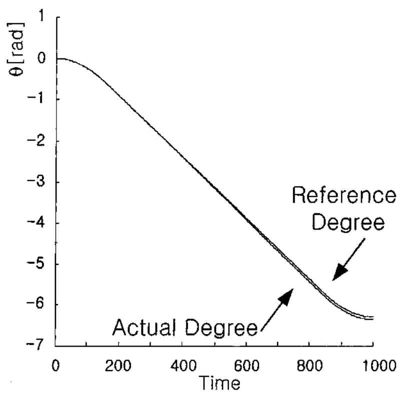

Fig. 10. Rotation angle of robot for round-shaped reference trajectory.

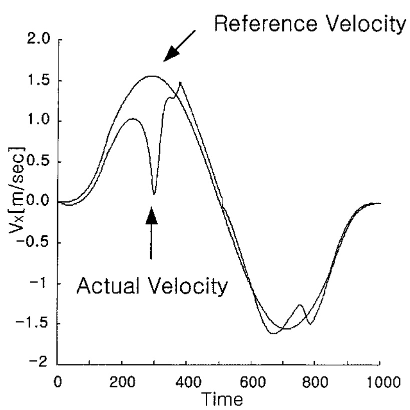

Fig. 11. X-velocity of robot for round-shaped reference trajectory.

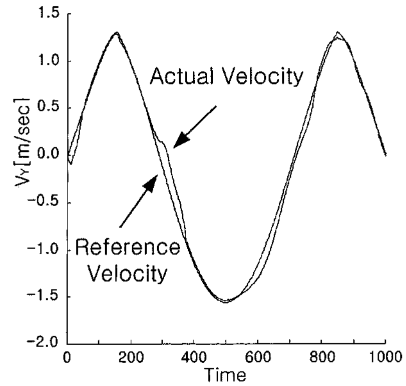

Fig. 12. Y velocity of robot for round-shaped reference trajectory.

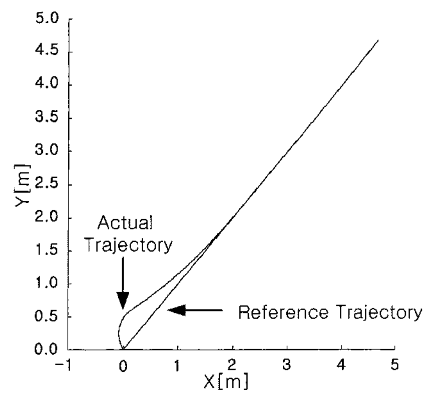

Fig. 13. Real trajectory of robot along diagonal reference trajectory.

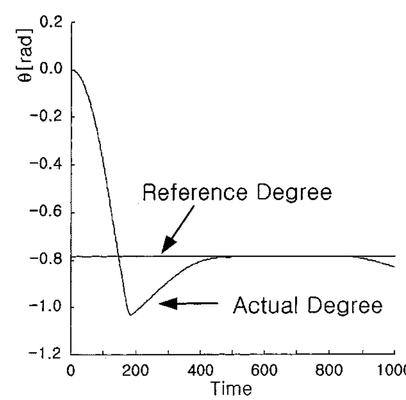

Fig. 14. Orientation angle of robot along diagonal reference trajectory.

The final value of the function is 1.15× bigger than the initial value of it in order to get gradual convergence

$$F(t) = \frac{1}{E(t)} \tag{7}$$

$$E(t) = \sum_{t=0}^{t=end} \left(a + b(t)\sqrt{e_p(t)^2} + c(t)\sqrt{e_{\dot{p}}(t)^2} \right)$$

where a is a constant weighting factor, and $b(t)$, $c(t)$ is a time-varying weighting factor. Results of computer simulations for round-shaped and diagonal reference trajectories are shown in Figs. 9–18. Simulation results for the round-shaped reference trajectory are shown in Figs. 9–12. The radius of the round-shaped reference trajectory is 2.725 m and the maximum rotation speed of the WMR is 0.57 rad/s. Total moving time of the WMR is 13 s. Acceleration and deceleration time of robot are 2 s each. The starting point position and finishing point position of the reference trajectory are $P_r = [-2.725,\ 0,\ 0°]$ and $Pr = [-2.725,\ 0,\ -360°]$, respectively. The starting point position of the robot is $P = [-1.5,\ 0,\ 0°]$.

Simulation of the round-shaped reference trajectory tracking shows a successful result. A drastic velocity drop is shown in the first half velocity profile along the X axis in Fig. 11. It is assumed that the robot reduces its velocity to change course in order to follow a predetermined trajectory. Initial rule base, searched input, and output membership functions are shown in Appendix A.

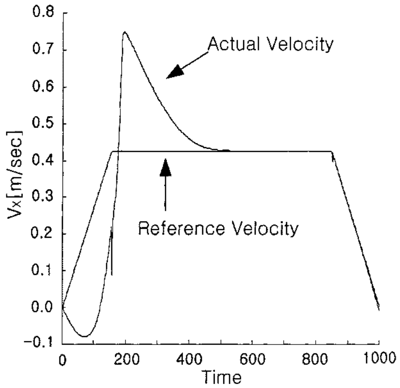

Fig. 15. Robot velocity of X axis along diagonal reference trajectory.

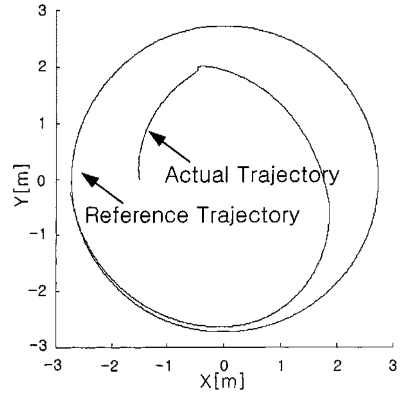

Fig. 17. Real trajectory of WMR for round-shaped reference trajectory without adaptation for output term set and rule base.

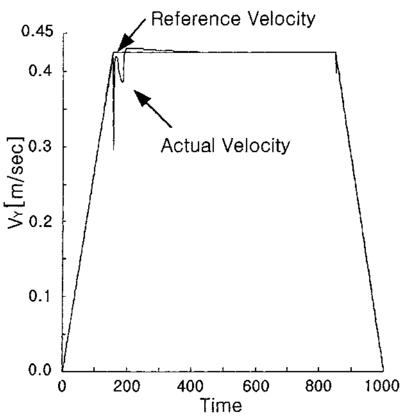

Fig. 16. Robot velocity of Y axis along diagonal reference trajectory.

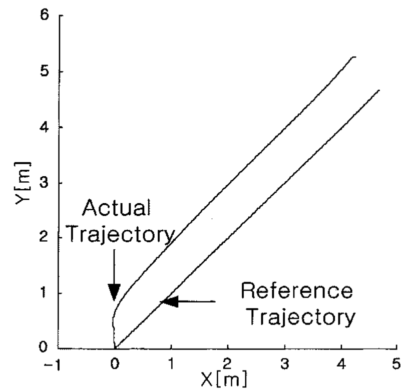

Fig. 18. Real trajectory of robot along diagonal reference trajectory without adaptation for output term set and rule base.

Simulation results for a diagonal reference trajectory are shown in Figs. 13–16. The full distance of the diagonal reference trajectory is 6.6 m and the maximum velocity of the WMR along the X and Y axes is 0.465 m/s. Total driving time is 13 s. Acceleration and deceleration times are 2 s each. The starting point position and finishing point position of the reference trajectory are $P_r = [0, 0, 45°]$ and $P_r = [4.67, 4.67, -45°]$, respectively. The starting point position of the robot is $P = [0, 0, 0°]$.

Simulation results for a diagonal reference trajectory show that the real robot trajectory appropriately converges to the reference trajectory. Initial negative velocity for the X axis is caused by robot rotation. This negative velocity is proportional to the difference between the center point of the robot's rotation and the point of the robot coordinate that is presumed. Sudden velocity ascent in the first half velocity profile occurs in order to remove accumulated error along the X axis. This error arose by the initial orientation difference between robot and predetermined trajectory. Initial rule base, searched input, and output membership functions are shown in Appendix B. Simulation results for the fuzzy controller without the increase of elements in output term set are shown in Figs. 17 and 18.

VI. Conclusion

In the conventional fuzzy-neural network model, the addition of new term nodes and regulation of the rule base are not easy. Moreover, this model produces a surplus node and it cannot take up general defuzzification method, even where the term node is appended. However, in the proposed algorithm, new term nodes can be appended according to the similarity measure. In addition, removal of surplus nodes and regulation of the rule base can also be carried out by learning from competition. The complexity of the system is reduced because the input variables in the antecedent clause are independent of each other and optimization becomes easier due to the reduction in complexity. Our self-organized fuzzy controller was designed and the feasibility of its control algorithm was tested by computer simulation. However, appropriate convergence balance at each step to find out an optimal rule base and optimal membership functions is necessary. If earlier convergence occurs at any step, then the proposed algorithm should lose its function, and appropriate selection of the chromosome $(A(a_{i\_\mathrm{final}}, b_{i\_\mathrm{final}}, c_{i\_\mathrm{final}}))$ is very important. Wrong selection of the chromosome may induce remarkable loss of effectiveness of the algorithm. For further study, an algorithm that is available in real time and is robust for chromosome $(A(a_{i\_\mathrm{final}}, b_{i\_\mathrm{final}}, c_{i\_\mathrm{final}}))$ should be designed.

TABLE II
INPUT MEMBERSHIP FUNCTIONS FOR ROUND-SHAPED REFERENCE TRAGECTORY

		NB	NS	ZE	PS	PB
e for X	center	−1.21	−1.08	0.00	0.18	0.41
	left	∞	0.87	1.91	0.03	0.09
	right	0.79	0.09	0.76	0.70	∞
e for Y	center	−1.00	−0.42	0.00	0.68	1.12
	left	∞	0.82	1.34	0.18	0.46
	right	0.24	0.02	0.60	1.67	∞
e for Θ	center	−1.19	−0.54	0.00	0.65	1.12
	left	∞	2.00	0.50	0.48	0.34
	right	0.48	0.38	0.93	1.28	∞
Δe for X	center	−1.09	−0.44	0.00	0.50	0.64
	left	∞	1.11	0.83	0.11	0.11
	right	0.41	0.17	0.94	0.70	∞
Δe for Y	center	−1.21	−0.09	0.00	0.39	0.57
	left	∞	0.74	0.52	0.04	0.15
	right	0.37	0.84	0.80	0.23	∞
Δe for Θ	center	−1.25	−0.64	0.00	1.47	1.82
	left	∞	1.40	0.37	1.13	1.58
	right	1.00	0.33	1.04	1.67	∞

TABLE III
OUTPUT MEMBERSHIP FUNCTIONS FOR ROUND-SHAPED REFERENCE TRAJECTORY

for X	center	−0.85	−0.39	0.00	0.12	0.37	0.47
	left	0.23	0.35	0.38	0.11	0.24	0.03
	right	0.31	0.32	0.33	0.08	0.43	0.11

for Y	center	−0.90	−0.71	−0.67	−0.24	0.00	0.33	1.00
	left	1.64	0.21	0.26	0.30	1.08	0.27	0.17
	right	0.03	0.28	0.29	0.01	0.46	0.64	0.94

for Θ	center	−0.51	−0.21	0.00	0.13	0.22	0.27	0.33
	left	0.21	0.14	0.24	0.06	0.19	0.14	0.17
	right	0.19	0.09	0.42	0.27	0.12	0.14	0.24

TABLE IV
INITIAL RULE BASE FOR X, Y, Θ

		error				
		NB	NS	ZE	PS	PB
Δerror	NB	NB	NB	NS	PS	PB
	NS	NB	NS	ZE	PS	PB
	ZE	NB	NS	ZE	PS	PB
	PS	NB	NS	ZE	PS	PB
	PB	NB	NS	PS	PB	PB

APPENDIX A

See Tables II–IV.

APPENDIX B

See Tables V–VII.

TABLE V
INPUT MEMBERSHIP FUNCTIONS FOR DIAGONAL REFERENCE TRAJECTORY

		NB	NS	ZE	PS	PB
e for X	center	−1.48	−0.56	0.00	0.65	1.33
	left	∞	0.74	0.70	0.12	1.01
	right	0.46	0.33	0.66	0.89	∞
e for Y	center	−0.82	−0.78	0.00	0.22	0.67
	left	∞	0.73	1.20	0.16	0.59
	right	0.85	0.48	0.31	0.24	∞
e for Θ	center	−0.74	−0.61	0.00	0.82	1.00
	left	∞	0.81	0.68	0.77	0.67
	right	0.41	0.18	0.75	0.97	∞
Δe for X	center	−1.29	−0.61	0.00	0.70	1.16
	left	∞	0.54	0.13	0.30	0.67
	right	0.53	0.56	0.75	1.07	∞
Δe for Y	center	−1.24	−0.73	0.00	0.25	0.60
	left	∞	0.23	0.02	0.35	1.34
	right	0.69	0.01	0.82	1.05	∞
Δe for Θ	center	−1.62	−0.96	0.00	0.59	1.81
	left	∞	0.74	0.83	0.55	0.63
	right	1.01	0.18	0.62	0.87	∞

TABLE VI
OUTPUT MEMBERSHIP FUNCTIONS FOR DIAGONAL REFERENCE TRAJECTORY

for X	center	−1.28	−0.91	−0.21	0.00	0.26	0.27	0.54	0.98
	left	0.43	0.29	0.21	0.42	0.23	0.19	0.39	0.76
	right	0.27	0.10	0.18	0.50	0.61	0.32	0.25	0.43

for Y	center	−1.49	−1.41	−0.37	0.00	0.67	0.89
	left	0.51	1.46	0.11	0.01	0.18	0.58
	right	0.33	1.36	0.24	0.10	0.30	0.23

for Θ	center	−0.36	−0.35	−0.22	0.00	0.12	0.30	0.35
	left	0.10	0.14	0.08	0.31	0.07	0.18	0.25
	right	0.03	0.27	0.07	0.22	0.09	0.17	0.18

TABLE VII
INITIAL RULE BASE FOR X, Y, Θ

		error				
		NB	NS	ZE	PS	PB
Δerror	NB	NB	NB	NS	PS	PB
	NS	NB	NS	ZE	PS	PB
	ZE	NB	NS	ZE	PS	PB
	PS	NB	NS	ZE	PS	PB
	PB	NB	NS	PS	PB	PB

REFERENCES

[1] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic Theory and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[2] D. Driankov, H. Hellendoorn, and M. Reinfrank, *An Introduction To Fuzzy Control*. New York: Springer-Verlag, 1996.
[3] M. M. Gupta and N. K. Sinha, *Intelligent Control Systems*. Piscataway, NJ: IEEE Press, 1996.
[4] J.-S. R. Jang and T.-C. Sun, *Neuro-Fuzzy and Softcomputing*. Englewood Cliffs, NJ: Prentice-Hall, 1997.
[5] G. V. Tan and X. Hu, “On design of fuzzy controller using genetic algorithm,” in *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, 1996, pp. 905–911.
[6] C.-C. Hsu, S.-I. Yamada, H. Fujikawa, and K. Shida, “A multi-operator self-tuning genetic algorithm for fuzzy control rule optimization,” in *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, 1996, pp. 842–847.

[7] J. L Castro, "How many rules are necessary to get a good fuzzy controller for control problem," in *Proc. 6th IEEE Int. Conf. Fuzzy Systems*, 1997, pp. 749–754.
[8] Y.-P. Hsu and C.-C. Tsai, "Autotuning for Fuzz-PI control using genetic algorithm," in *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, 1996, pp. 602–607.
[9] C. T. Lin, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Comput.*, vol. 40, pp. 1320–1336, Sept. 1991.
[10] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin, Germany: Springer-Verlag, 1995.
[11] Y.-O. Chung and C. Park, "Position, orientation, and velocity feedback control algorithm for differential mobile robot," *J. Korean Inst. Elect. Eng.*, vol. 34, no. 11-S, pp. 63–72, 1997.

Sung Hoe Kim was born in Seoul, Korea, in 1969. He received the B.S. and M.S. degrees in electronic engineering in 1993 and 1995, respectively, from Kyung Hee University, Seoul, Korea, where he is currently working toward the Ph.D. degree.

His research interests include control theory, neural networks, fuzzy logic, and microprocessor applications.

Chongkug Park (M'89) was born in Korea in 1945. He received the B.S. degree in physics from Seoul National University, Seoul, Korea, and the M.S. and Ph.D. degrees in electrical engineering from Yonsei University, Seoul, Korea, in 1971, 1975, and 1979, respectively.

From 1987 to 1988, he was a Visiting Professor at Oregon State University. From 1995 to 1998, he was Associate Dean, Graduate School of Industry and Information, Kyung Hee University, Seoul, Korea, where he is currently a Professor and Dean, College of Engineering. He was President of the the Korea Fuzzy Logic and Intelligent Systems Society (KFIS) from 1998 to 1999. He was a Member of the Board of Directors of the Society of Instrument and Control Engineers of Japan from 1999 to 2000. His research interests are robotics, mechatronics, fuzzy logic, and neural networks.

Prof. Park received the Academic Achievement Award from the Korea Institute of Electrical Engineers (KIEE) in 1979 and Distinguished Service Awards from the Institute of Electronics Engineers in Korea (IEEK) and KFIS in 1996 and 2000, respectively.

Fumio Harashima (M'75–SM'81–F'88) was born in Tokyo, Japan, in 1940. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1962, 1964, and 1967, respectively.

He was an Associate Professor in the Institute of Industrial Science, University of Tokyo, in 1967, and a Professor from 1980 to 1998. He was Director of the Institute from 1992 to 1995. Since April 1, 1998, he has been President of Tokyo Metropolitan Institute of Technology, Tokyo, Japan. He is also currently Professor Emeritus of the University of Tokyo. His research interests are in power electronics, mechatronics, and robotics. He is the coauthor of four books and has authored more than 1000 published technical papers. He has been active in various academic societies, such as the Institute of Electrical Engineers of Japan (IEEJ), Society of Instrument and Control Engineers of Japan (SICE), and Robotics Society of Japan. He is currently President-Elect of the IEEJ.

Dr. Harashima served as President of the IEEE Industrial Electronics Society (IES) in 1986–1987 and as Secretary in 1990. He was a member of the IEEE Executive Committee and Board of Directors in 1990. He was also a member of the IEEE N&A Committee in 1991–1992 and IEEE Fellows Committee in 1991–1993. In 1995, he served as the Founding Editor-in-Chief of the IEEE/ASME TRANSACTIONS ON MECHATRONICS. He has received a number of awards, including the 1978 SICE Best Paper Award, 1983 IEEJ Best Paper Award, 1984 IES Anthony J. Hornfeck Award, 1988 IES Eugene Mittelmann Award, and IEEE Millennium Medal in 2000. He is a Fellow of SICE.