Paper

Adaptive Fuzzy Control Based on Neural Network for a Mobile Vehicle

Masanori Sugisaka* Member
Fengzhi Dai* Non-member

This paper presents a practical control method for the experimental mobile vehicle. By merging the advantages of the neural network, adaptive algorithm and fuzzy control, the adaptive fuzzy control based on neural network is presented. This adaptive fuzzy control system can deal with a large amount of training data by the neural network, from these data produce more reasonable fuzzy rules by the adaptive (clustering) algorithm, at last control the object by the fuzzy control. It is not the simple combination of the three methods, but merging them into one control system. Experiments and some future considerations are also given.

Keywords : hybrid system, neural network, clustering algorithm, fuzzy control, adaptive fuzzy control, mobile vehicle

1. Introduction

Traditional control systems are always based on mathematical models, in which they are described by continuous or differential equations that can define the system's response to their inputs. In many cases, the mathematical models of the control process may not exist, or may be too difficult to get them.

Neural network (NN) control is one of the new and exciting control fields. The neural network control is different from the conventional control methods. It has the adaptive and self-learning ability for complex and uncertainty problems. It can deal with a large amount of training data, and from which the relationships among them can be found, although unrelated information and noise are in them [(1)].

Adaptive control is a method that designs the controller, estimates the object or controller's parameters in on-line fashion, and presents outputs to control the object. There are two ways in the field of the adaptive control. One is the indirect design for the adaptive controller; the other is the direct adaptive controller design. The difference between them is whether the model of the object should be obtained before or when it is being controlled [(2)].

Also, in a system, the values of variables may be within a range of states. Sometimes the transition from one state to another is hard to define. The way to solve it is to make the states "fuzzy": allow them to change gradually from one state to another [(3)]. The general feature of the fuzzy control is: all the rules that apply are invoked, using the membership functions and true values obtained from the inputs, to determine the result of the rules.

Each control method has its advantages and disadvantages. The paper wants to present the application to merge them, so that the advantages will be adopted while the disadvantages eliminated [(4), (5)]. It is a mechanism, by which the control system can be utilized from the expert knowledge, to solve a special problem.

1.1 Hybrid systems

When more than one problem-solving technique (neural networks, fuzzy logic, genetic algorithms, expert systems, artificial life, etc) are used in order to solve a problem, they are called hybrid systems. There are three main structures for them to be connected into one hybrid system shown in Table 1 [(6)].

Table 1. Three main structures of hybrid systems and examples.

1 Sequential Hybrid	The first technique passes its output to the second. It is the weakest form of hybridization
	Statistical pre-processor that passes its output (based on factor analysis) to a neural network.
2. Auxiliary Hybrid	The second technique is called by the first one and returns some information. It is intimately involved with the first, while the first can exist along.
	Neural network that calls a genetic algorithm module to optimize its structure.
3. Embedded Hybrid	The first technique contains the second one. Neither technique can be defined without the other.
	Neural network-fuzzy logic hybrid (to simulate structure and features of the fuzzy system).

We have to point out that: although hybrid systems are intrinsically better, which allow for the synergistic combination of two techniques with more strength and less weaknesses than either alone, they have more chances for misuse than single technique [(6)]. It should be considered when construct a new hybrid system, which will be described in section 3.

The experimental object in the lab is an intelligent mobile vehicle (MV). An autonomous MV or car-like robot can be used in various fields. For example, a human-assistant mobile vehicle can help a physically handicapped person. A nursing robot can be used in the hospital, and an intelligent mobile vehicle or robot can work in dangerous, such as nuclear reactor environment, etc.

There are many methods and applications have been presented to control MV or robots. K. Watanabe et al [(7)] and K. Izumi et al [(8)] presented the fuzzy behavior-based system for a mobile robot. The control is divided into different behavior-based functions that are based on the different recognized objects. It is convenient to add new behavior functions into the system, but for the stored functions, they are difficult to be adjusted based on the changeable environment. H. Wang et al [(9)] developed a fuzzy logic Kalman filter estimation method for a two-wheel steerable vehicle. This method merged the fuzzy control and Kalman filter, but it is not an

* Department of Electrical and Electronic Engineering, Oita University, Dannoharu 700, Oita, Japan, 870-1124

intelligent control system. X. Wang [10] and M. Sugisaka et al [11] presented an artificial brain theory for a mobile vehicle. They are not without their limitations. The problem is the conflict between stability and plasticity:

If a control system can adapt to the change of the environment, it is called plastic. And stability means that the system preserves previously learned knowledge. Based on the change of the environment and the system states, how to adjust the controller or system's parameters adaptively without destroying the trained control rules is important.

Conventional NN control cannot overcome this problem. Too often, learning a new pattern has to modify previously trained weights. And adaptive control needs to estimate the system parameters. Also reasonable fuzzy rules are difficult to be obtained if only based on experience, trial and error. The fusion of them named adaptive fuzzy control based on neural network (AFCNN) is an effective solution. The adaptive fuzzy control system will be discussed in section 3. It is a kind of directly adaptive controllers, and belongs to the third structure in Table 1. From the experimental results (in section 4.3 and 4.4), based on the change of the environment, fuzzy rules can be optimized and updated from the candidates of them.

1.2 AFCNN hybrids The data-driven learning technique (the neural network) and knowledge-driven technique (the fuzzy system) will be considered in this paper. The integration of NN and the fuzzy control is to produce a new field called the neuro-fuzzy system. In fact, neuro-fuzzy logic hybrids are based on using NN to simulate a fuzzy system, or to adjust the membership functions.

AFCNN belongs to the neuro-fuzzy logic hybrids. It is capable of extracting and optimizing fuzzy rules and sets from the input-output training data, which is the first time to combine them into one system. The purpose of the experiment is to make the MV system follow the guideline accurately. The clustering methods are used to produce fuzzy rules for the AFCNN system. And by clustering techniques to mine the important input-output relationship from the input-output data pairs to form and optimize the fuzzy rules and sets is the main thought of the paper.

In this paper, section 2 gives the general configuration of the experimental mobile vehicle. In section 3, the fuzzy rule producer and AFCNN system are described. There are two main parts in section 3. The fuzzy rule producer is used to examine the reliability of the fusion among NN, the adaptive algorithm and fuzzy control. The other is that by AFCNN, the mobile vehicle is experimented, which is the main part of the paper.

Section 4 gives the experimental results and analysis. After the introduction of the former control method to MV, section 4.3 gives the result of the fuzzy rule optimization. And section 4.4 is the result of using clustering algorithm to adapt to some parameters (the center and width of each membership function) of the fuzzy sets. In section 5, conclusions are drawn.

The object of the paper is to present an effective and practical control method, which can be used to obtain the reasonable fuzzy control rules adaptively by NN.

2. Configuration of the Mobile Vehicle

Fig.1 gives the photo and the mechanical configuration of the experimental MV [10]–[12]. It is 1.5m×0.8m×0.4m and 150kg.

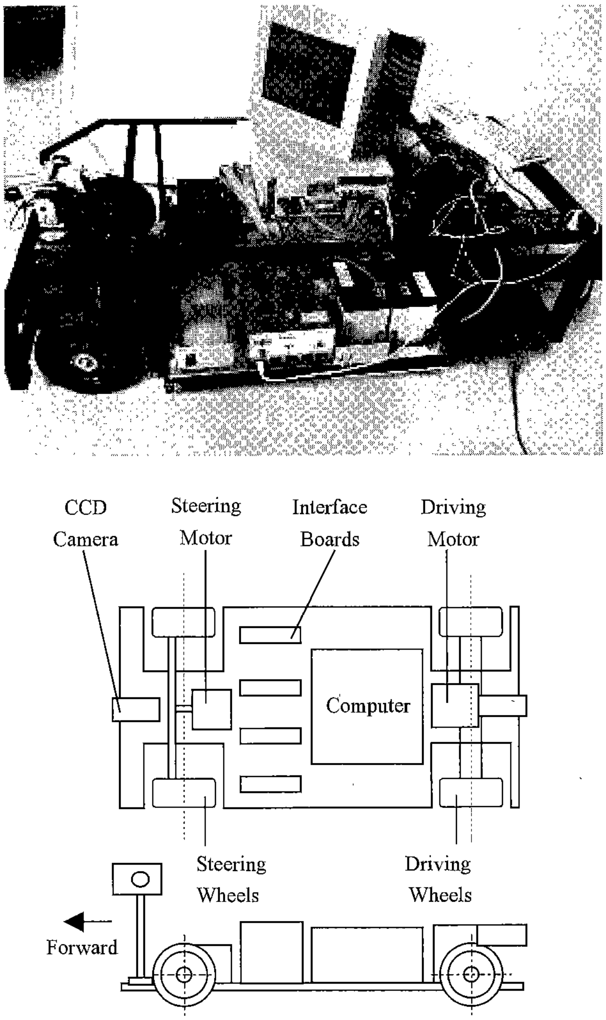


Fig. 1. Mechanical configuration of MV.

The MV system is comprised of a rigid metal frame to support the equipment, a CCD camera and digitizer, I/O systems and power (the battery system) supplies that enable completely autonomous operation, and a computer is the control center.

From the motion control perspective, MV has two degrees of freedom: velocity and angular velocity (realized by the driving and steering motors, respectively). The experimental MV has one DC stepping motor to drive MV rear wheels to move forward or backward, as well as one stepping motor to steer the front wheels to rotate left or right. An encoder is used to detect the real rotation angle of the front wheels. The sensor for image processing is a CCD camera. Thus, the sensors are the CCD camera and the encoder. The input data to the computer are the result of image processing and the output of the encoder. The control parameters are the duty ratio and the number of pulses inputted to those motors.

MV is the experimental object. The goal of the control is to make MV move on the road tracking the guideline smoothly.

3. Structure of AFCNN

Many control problem can be involved into decrease the input error variable and the derived change in error. By using the fuzzy logic, the inference from experience can be obtained. And NN is used to cluster the training data by the adaptive algorithm, so that the fuzzy rules and sets can be optimized. That is the principle of the AFCNN system.

The individual structure and algorithm of NN and fuzzy control will not be discussed in the paper. This section gives the structure of AFCNN. First, a fuzzy rule producer will be considered.

3.1 Fuzzy Rule Producer Generally, there are three types of fuzzy models. Eq.(1) shows the type 1 and 2, Eq.(2) gives the type 3. The general fuzzy controller is always to be used to deal with fuzzy relationship between the input and output states. For example, the fuzzy rules for a two-input one output system are

$$\begin{cases} \text{Rule1 : If } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{, then } z \text{ is } C_1, \\ \text{Rule2 : If } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{, then } z \text{ is } C_2, \\ \qquad \cdots \quad \cdots \quad \cdots \\ \text{Rulem : If } x \text{ is } A_m \text{ and } y \text{ is } B_m \text{, then } z \text{ is } C_m. \end{cases} \quad \cdots\cdots (1)$$

Here x, y and z are fuzzy variables. A_i, B_i (i = 1, 2, …, m) are fuzzy values that belong to {NB, NM, NS, ZE, PS, PM, PB}, which can be described by membership functions. NB, ZE, PS, … are fuzzy terms that representing negative big, zero, positive small, …, respectively. If the output C_i is also given by a fuzzy value, it is type 1. If C_i is a singleton constituent, it is type 2.

And if the output C_i is the function of the input data, it is type 3. Takagi and Sugeno fuzzy algorithm [5], [13] can deal with this kind of fuzzy models and then accurate output from fuzzy rules can be obtained. Fig.2 gives its structure, in which only two fuzzy rules are given for simply explaining.

Define $C_i = f_i(x, y)$, the fuzzy rules are:

$$\begin{cases} \text{Rule1 : If } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{, then } z \text{ is } f_1(x, y) \\ \text{Rule2 : If } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{, then } z \text{ is } f_2(x, y) \\ \qquad \cdots \quad \cdots \quad \cdots \\ \text{Rulem : If } x \text{ is } A_m \text{ and } y \text{ is } B_m \text{, then } z \text{ is } f_m(x, y) \end{cases} \quad \cdots\cdots (2)$$

If an input set (x_0, y_0) is presented, the combined value ω_i of each fuzzy rule is:

$$\begin{cases} \omega_1 = \min\{A_1(x_0), B_1(y_0)\} \\ \omega_2 = \min\{A_2(x_0), B_2(y_0)\} \\ \qquad \cdots \quad \cdots \quad \cdots \\ \omega_m = \min\{A_m(x_0), B_m(y_0)\} \end{cases} \quad \cdots\cdots (3)$$

Define ω_i^0 as the normalized value of ω_i (i = 1, 2, …, m),

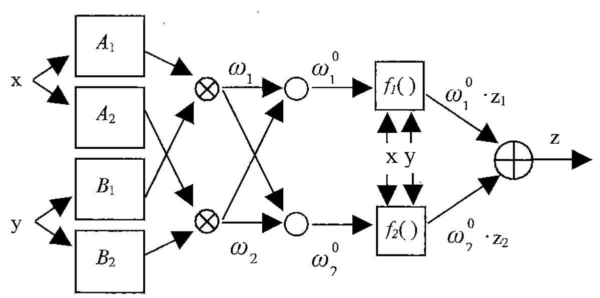


Fig. 2. Fuzzy rule producer.

$$\omega_i^0 = \frac{\omega_i}{\sum_{k=1}^{m} \omega_k} \quad \cdots\cdots (4)$$

Then the output z is

$$z = \omega_1^0 z_1 + \omega_2^0 z_2 + \cdots + \omega_m^0 z_m$$

$$= \frac{\sum_{k=1}^{m} \omega_k f_k(x_0, y_0)}{\sum_{k=1}^{m} \omega_k} \quad \cdots\cdots (5)$$

where z_i is the result of each function $f_i(x, y)$.

Two input variables x, y, and one output z in the fuzzy rule are true values. By adjusting the membership functions [13], the function between two inputs and one output can be approximated.

3.2 AFCNN Section 3.1 shows that from the input-output sets, the complicated relationship between inputs and outputs can be simulated by fuzzy rules. Now the problem is: faced to a large amount of the given input-output training sets, how to get the fuzzy rules adaptively. AFCNN can solve this problem, which is given in Fig.3. And section 4.3 and 4.4 give the results for optimization of fuzzy rules and updating of fuzzy membership functions, respectively.

In fuzzy set theory, the element belongs to the set partially with a certain degree. All reasoning rules are used in parallel, and the results are combined according to the fuzzy rules. Before the fuzzy sets can be updated, the fuzzy rules have to exist. And only when needed should a new fuzzy rule be created [14]:

(1) The errors are removed in descending order;
(2) New fuzzy rule is not added before the existing fuzzy rules have been turned;
(3) The existing fuzzy rules must not be destroyed.

To form and adjust fuzzy rules, first, the centers of membership functions are founded by dividing each input and output linguistic variable into predetermined number of fuzzy values. The widths of fuzzy sets are initially determined by nearest neighbor or k-means heuristics [14]. After that, the fuzzy rules are founded by feeding training data set through AFCNN and deciding which fuzzy rules are the most important. Some similar consequences can be combined into one fuzzy rule. Finally, the centers and widths of the fuzzy sets are turned using a gradient descent algorithm.

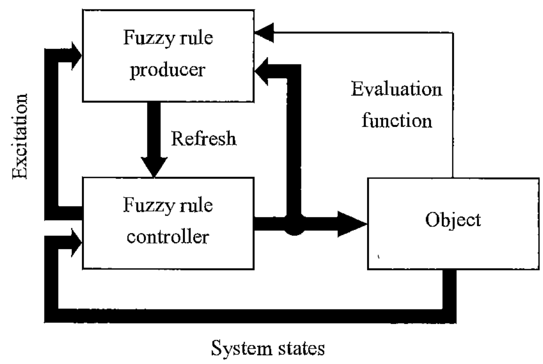


Fig. 3. Structure of AFCNN.

Usually, there are two kinds of adaptive algorithms to cluster the training data. They are the growing partitioning algorithm (GPA) and shrinking partitioning algorithm (SPA) [15]. GPAs start from empty cluster and adapt new clusters until a pre-selected scale is satisfied. For example, the adaptive resonance theory (ART) neural network is one of GPAs, which has been used as pattern recognizer [1], [16]. SPAs start with a reasonable number of clusters and shrink the number of clusters by certain algorithm (for example, the following introduced competitive learning adaptive vector quantization (CL-AVQ) algorithm). AFCNN belongs to SPAs and is divided into the following four steps (the theory is given in Appendix):

(1) Define the fuzzy universe of discourse for both input and output variables (the number and centers for fuzzy rules). For example, input fuzzy sets are $\{I_1, I_2, \ldots, I_r\}$, outputs are $\{O_1, O_2, \ldots, O_s\}$. I_i (i = 1, …, r) and O_j (j = 1, …, s) belong to fuzzy values {NB, NM, NS, ZE, PS, PM, PB}. There are r×s possible fuzzy rules (candidates) to describe the rule of $I_i \rightarrow O_j$. In Table 2, r = 7, s = 7.

(2) Define maximum and minimum values for each I_i and O_j, thus the upper and lower bounds (width) of each fuzzy variable for possible fuzzy control rule are defined. Then the given data (true values) can be transformed to fuzzy values. Table 2 shows the distribution of fuzzy sets.

(3) Establish a no-hidden layer feedforward NN shown in Fig.4. The input and output nodes are NI_i and NO_j, the character N is used to distinguish them from the inputs and outputs of fuzzy control. The number of input nodes p is the sum of the input and output variables of the fuzzy controller. The number of output nodes q is multiplication of the numbers of input and output fuzzy sets. For example, in Table 2 the input nodes for NN are two (one input and one output) and the output nodes are 7×7 = 49. In Fig.2, the fuzzy variables are x, y, and z, then the input nodes for NN is three. And if each of the variables is divided into five fuzzy values, the output nodes are 5×5×5 = 125.

(4) By the CL-AVQ algorithm created in this paper, the neural network is trained. After that, calculating the number of training data sets that falling into each grid in Table 2, those have more numbers are the selected fuzzy rules.

Table 2. Distribution of fuzzy sets.

I_i \ O_j	NB	NM	NS	ZE	PS	PM	PB
NB							
NM							
NS							
ZE							
PS							
PM							
PB							

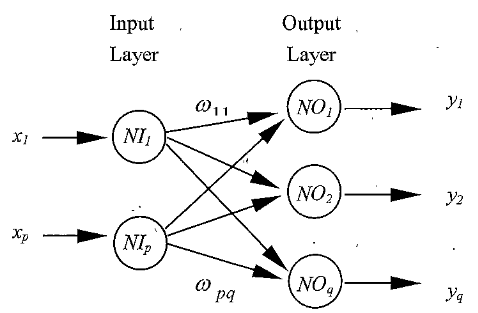


Fig. 4. No-hidden layer feedforward NN.

This method has been proved that it has the solution and can be converged to it [17]. By training the neural network, the clustering algorithm can classify the training data into several well-optimized fuzzy rules.

3.3 CL-AVQ Algorithm The learning vector quantization (LVQ) algorithm is a supervised training method for classification [18]. The goal of it is to move the decision border between nodes towards the Bayesian limit so that the number of misclassification is minimized (suggested by section 4.4). And the adaptive vector quantization (AVQ), which is used to train NN of Fig.4, divides the input and output spaces into overlapping fuzzy sets. After that, the fuzzy rules are found out through the best relations between input and output fuzzy sets. Its performance is partly determined by the initial partitioning of the input and output spaces (that will be considered in Appendix).

Competitive learning is described by the differential equation

$$\dot{\omega}_{ij} = S(y_j)(x_i - \omega_{ij}) \quad \cdots\cdots(6)$$

$\dot{\omega}_{ij}$ is the differentiation of ω_{ij}. And $S(\cdot)$ is a competitive function

$$S(y_j) = \frac{1}{1 + e^{-cy_j}} \quad \cdots\cdots(7)$$

$$y_j = \sum_i x_i \omega_{ij} \quad \cdots\cdots(8)$$

x_i is the value to the input layer node. ω_{ij} is the weight that connects between the input node i and the output node j. $S(\cdot)$ likes the step function according to c>>1. If $S(y_j)=1$, the output node j wins the competition, otherwise $S(y_j)=0$.

$S(y_j)=1$ will be satisfied only when the input vector has the minimal distance with the weights to the output node j:

$$\min_j \{\sum_{i=1}^{p} (\omega_{ij} - x_i)^2\}^{1/2} \quad \cdots\cdots(9)$$

where p is the number of the input nodes in Fig.4.

Replacing the differentiation in Eq.(6) by difference, namely, $\Delta\omega_{ij} = \omega_{ij}(k+1) - \omega_{ij}(k)$, the weights between the input nodes NI_i (i = 1, …, p) and the output node NO_j will be updated by

$$\omega_{ij}(k+1) = \omega_{ij}(k) + S(y_j)(x_i - \omega_{ij}(k)) \quad \cdots\cdots(10)$$

For those output nodes that fail the competition, the weights will not be modified.

Eq.(10) can also be written by

$$\omega_{ij}(k+1) = \omega_{ij}(k)(1 - S(y_j)) + x_i S(y_j) \quad \cdots\cdots\cdots\cdots (11)$$

Eq.(11) gives the following reality: the value $\omega_{ij}(k+1)$ is composed by two parts. One is that the strength of old knowledge $\omega_{ij}(k)$ is declined. The other is the strength of new knowledge x_i is increased. The changes occur within one training data set and both of them are changed exponentially. It can also be considered that $\omega_{ij}(k+1)$ moves to x_i at the exponential speed.

Thus the steps of CL-AVQ algorithm used to train NN are:

(1) Preparing. Determining the fuzzy feature space and the structure of NN.

(2) Initialization. Initializing weights ω_{ij} by random function.

(3) Competition. Finding out the minimal value in Eq.(9) from all the output nodes. The output node j is the winner.

(4) Based on competitive learning, adjusting all the weights to the output node NO_j by Eq.(10).

(5) Repeating steps (3) and (4) until all the data are trained.

4. Experiment

During the experiment, MV is driven to track as closely as possible to the guideline that was located on the ground. The former research of control in our lab will be simply introduced below at a level that the general idea of it can be gotten.

4.1 Former Methods for Experimental MV An algorithm called the three-center method (shown in Fig.5) has been introduced for image processing in reference (10).

The presented three-center method is that the image of a target is divided into three parts in the vertical direction. The location and the direction of the guideline in an image are determined by three centers of gravity (g_1(t), g_2(t) and g_3(t)) instead of only one. From these three central points, an overall position of the target g(t) = { $\xi(t)$, $\eta(t)$} with the parameters of the reference angle $\beta(t)$ (for the front wheels rotating) and the tendency of direction is obtained (shown in Fig.6).

Also, in order to keep visual tracking system running, learning method called the re-classification algorithm was presented [10]. The strategy is to let the mobile vehicle track the guideline by using the stored control arrays, from the cost function to adjust the past control rules to get more accurate patterns, and then the new control array for the guideline can be trained. Thus the strategy is by adjusting the stored control patterns to get more accurate ones.

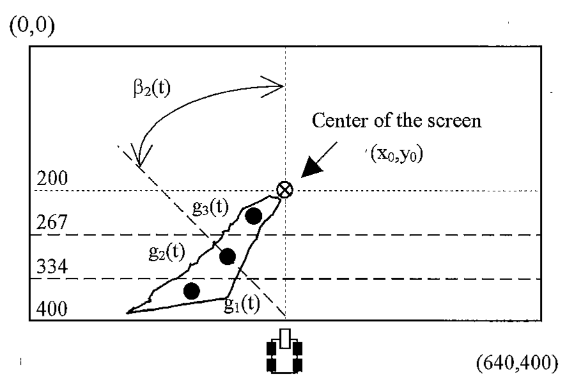


Fig. 5. Three-center method.

One obvious weakness of the method is that it is only suitable for tracking relatively simple or fixed trajectories.

For comparison, the same guideline is used in this paper to compare the control effect with the general fuzzy control presented in reference (10).

4.2 Experimental Object and Fuzzification The structure of MV has been given in section 2. And Fig.6 gives the scene taken by the CCD camera, from which the control parameters can be obtained.

The screen is divided into 640×400 pixels. The center of the screen is also the center of the camera. By image processing and the three-center method, the center of gravity of the target (ξ, η) can be obtained. Thus the parameter β (t), which represents the angle between the central line of MV and the target, should be converted to the number of pulses and then input to the steering motor to steer the front wheels rotating left or right to head to the center (ξ, η).

As introduced in section 2, MV has two degrees of freedom - velocity and angular velocity control. From the driver's point of view, they are mobility and steeribility. This paper gives the experimental result for the steering control. The input variables for the fuzzy control are β (t) and $\Delta\beta$ (t), where $\Delta\beta$ (t) = β (t) - β (t-1). The output of the fuzzy controller is the relative strength ϕ (t) used to control the steering motor.

Assuming that there is little knowledge about the fuzzy rules for the steering control. Based on experience and some presented results [10], Table 3 gives the assumptive fuzzy rules (candidates), which presents all the 25 possible rules.

β (t), $\Delta\beta$ (t) and ϕ (t) are all divided into five fuzzy values {NM, NS, ZE, PS, PM}. N in fuzzy control means negative, here in β (t) means the target is on the left side of the screen, in $\Delta\beta$ (t) means the target is located more left at time t than time t-1, and in ϕ (t) will let the front wheels turn left. P (positive) is opposite to N. Fig.7 gives the initial membership functions of those fuzzy

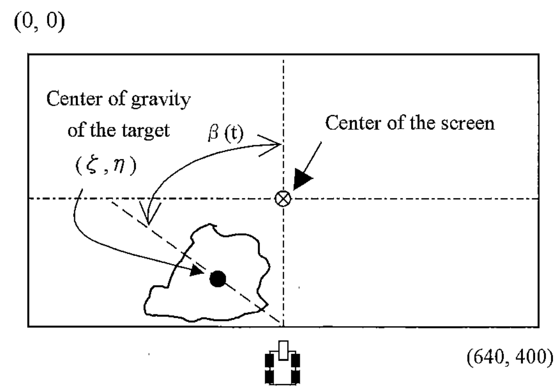


Fig. 6. Image processing method.

Table 3. All possible fuzzy rules for ϕ (t).

	NM	NS	ZE	PS	PM
NM	NM	NM	NM	ZE	PS
NS	NM	NM	NS	ZE	PS
ZE	NM	NS	ZE	PS	PM
PS	NS	ZE	PS	PM	PM
PM	NS	ZE	PM	PM	PM

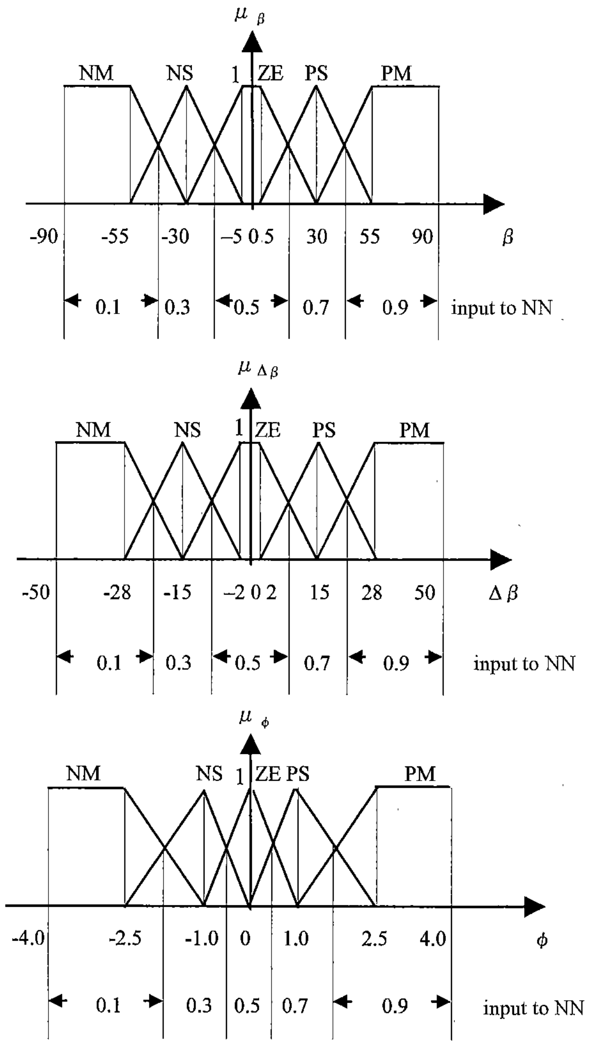
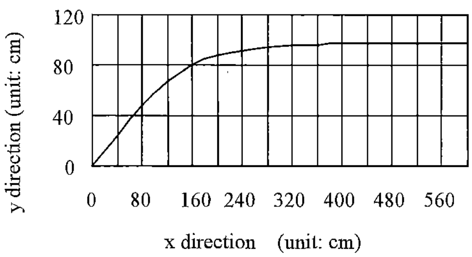
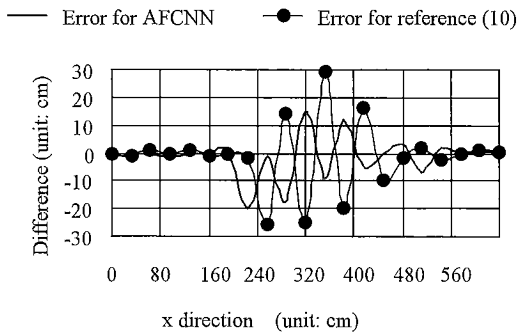


Fig. 7. Fuzzy membership functions.



(a) The experimental curve



(b) Experiment results

Fig. 8. Experiment results of AFCNN.

variables.

4.3 Experiment for Fuzzy Rule Optimization For comparison, the same guideline is presented to MV [(10)]. The sets ($\beta(t_k)$, $\Delta\beta(t_k)$, $\phi(t_k)$) are the real fuzzy control data (initially by Fig.7) sampled from experiments based on Table 3, which has 25 candidate fuzzy rules. They are not used as trial and error, but as the inputs to the no-hidden layer feedforward NN shown in Fig.4. The aim of the experiment in this section is to show how to optimize the fuzzy rules based on AFCNN.

The training process is to input these training data sets into the input layer of NN one by one, as introduced in section 3.2. The output layer node j who satisfies Eq.(9) is the winner, and all the weights connect to it will be adjusted by the CL-AVQ algorithm that had been introduced in section 3.3. Since these data are gotten when MV tracks the guideline, when all the data are trained, the procedure is over.

Fig.8 is the comparison of the experimental results coming from reference (10) and by AFCNN. The same guideline in Fig.8(a) is used. To simulate the change of the environment, the guideline is divided into three parts. The first one is a straight line from 0 to 120cm in the x direction, the second one is a curve from 120cm to 320cm, and the last part is a horizontal line.

Compared to the general fuzzy control in reference (10), in which the fuzzy rules and sets were gotten by experience, trial and error, the control result is given by Eq.(12) and Eq.(13):

$$\text{effect} = \frac{\sum_{i=1}^{n} |error|}{n} \qquad \cdots\cdots(12)$$

$$\text{similarity} = \frac{\sum |error_q - error_p|}{\sum |error_q|} \qquad \cdots\cdots(13)$$

In Fig.8, *error* is the difference between the sampled value and the curve of the guideline, and n is the number of the sampled data. Thus Eq.(12) gives the control result: the more the value of *effect* is small, the more the control is good. It is also called as average absolute error (AAE) equation. In Eq.(13), $error_q$ and $error_p$ represent two kinds of errors controlled by the general fuzzy control and AFCNN, respectively. Thus it gives the similarity of two control methods: the more the value of *similarity* is small, the more the control by AFCNN is similar to the general fuzzy control.

For the first part of the guideline, $effect_q^{(1)} = 2.23$, $effect_p^{(1)} = 2.11$, and $similarity^{(1)} = 0.06$. Similarly, $effect_q^{(2)} = 21.03$, $effect_p^{(2)} = 11.60$, and $similarity^{(2)} = 0.36$. And $effect_q^{(3)} = 3.53$, $effect_p^{(3)} = 3.36$, and $similarity^{(3)} = 0.07$.

Table 3 shows that the number of the candidate fuzzy control rules is 25, which are mostly gotten by experience, trial and error. The training data distributed in those 25 grids are given in Table 4.

Table 5 is the results after training, which shows that only 13 rules are adopted. The number of data falling into these 13 grids exceeds to a pre-determined value, so they are selected. The

Table 4. Distribution of training data.

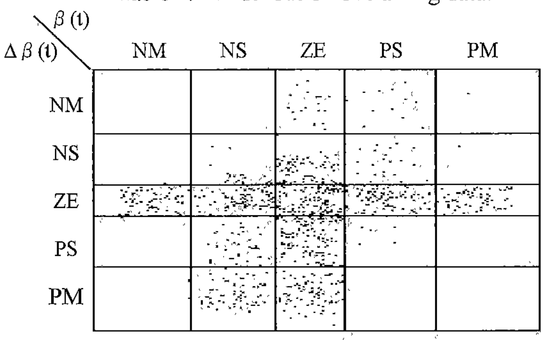


Table 5. Result fuzzy rules for ϕ (t).

β (t) \ $\Delta\beta$ (t)	NM	NS	ZE	PS	PM
NM			NM	ZE	
NS			NS	ZE	
ZE	NM	NS	ZE	PS	PM
PS		ZE	PS		
PM		ZE	PM		

strengths of others are so small that they can be neglected (One thing is that based on the fuzzy theory, some control rules are so important that even the numbers of them are small, they should be taken into account, see section 4.5).

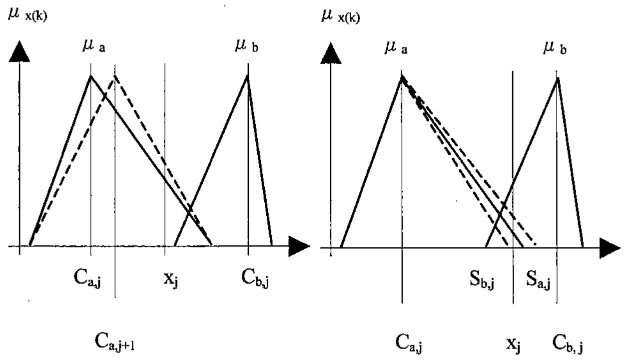
The reason why the fuzzy rules are optimized is that, in Table 3, all the possible control rules are listed, some of them are invalid, and some of them cannot be reached during the experiments (but they may be difficult to be found if only by experience). For example, the top-left fuzzy rule in Table 3 is:

$$\text{If } \beta \text{ (t) is NM and } \Delta\beta \text{ (t) is NM, then output } \phi \text{ (t) is NM} \quad \cdots\cdots(14)$$

It means that at time t, the center of gravity of the target is on the most left side of MV, and it is more left than at time t-1. This can only occur under the following two conditions. One is the control rule at time t-1 was wrong, the other is that the trajectory of the target or guideline varies exceeding the control limit of MV. The first one will not occur if the rules have their rationality. And if assuming that MV can track the trajectory of the target, which is a physically realizable system, the second problem can also be solved. Thus the fuzzy rule of Eq.(14) can be excluded easily.

But before training, or other words, if we have little information about the control object, some fuzzy rules may be difficult to be found their irrationalities (includes the determination of the fuzzy rules, and the center and width of each membership function). They may exist as fuzzy rules in the control system that will never be reached. By AFCNN, they can be shown that there is few data belonging to them, and then we can analyze the reason and modify the fuzzy rules to be more reasonable.

From Fig.8(b), after trained by AFCNN, when the guideline was a straight line, the control results were almost the same. When the guideline became complicated, control rules by AFCNN were better because they were optimized: the control rules are decreased effectively, so that the control time and program can be improved greatly. From Fig.10 (in section 4.4), we also see that the widths of edge grids become larger than the initial settings in



(a) Center updating (b) Spread updating

Fig. 9. Updating the membership function.

Fig.7 so as to get stronger control results. Thus the errors of control became smaller, especially when the guideline was complicated.

Fig.8 and Table 5 only show that AFCNN has better control result than that of only by the fuzzy control compared to reference (10), and can obtain more reasonable fuzzy rules from candidates based on the number of each fuzzy rule adopted. But they can say little about the feature of AFCNN to update parameters of the membership functions, which will be given in the next section.

4.4 Updating of Fuzzy Membership Function Optimizing of the membership function includes adjusting both the center and the spread of each fuzzy value. After the fuzzy rules are determined, tuning the membership functions should be considered. The classification decision of LVQ2.1 [19] is used in the paper, which is based on the idea of differentially shifting the decision borders towards the Bayes limits.

From Table 2 to Table 5, it seems that all the fuzzy sets are adjacent, but in fact, they are set overlapped (see Fig.7). The form shown in tables is drawn only for easy explanation.

If the input-output fuzzy set fires only one value of fuzzy membership function by x_j, the center of that function should be updated (Fig.9(a)). When x_j fires two adjacent membership functions simultaneously, one of them is the winner, and the spread of the other membership function will be moved to one of the dot lines (Fig.9(b)).

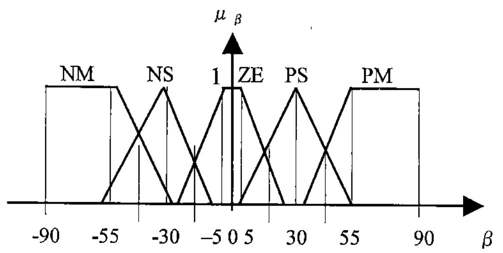
In Fig.9, C_a, C_b, S_a and S_b are central and spread points for adjacent fuzzy membership functions μ_a and μ_b. x_j is the value of the fuzzy variable. For Fig.9(a), x_j only fires μ_a, the center of the fuzzy membership function μ_a is updated by

$$C_{a,j+1} = C_{a,j} + \delta(x_j - C_{a,j}) \quad \cdots\cdots(15)$$

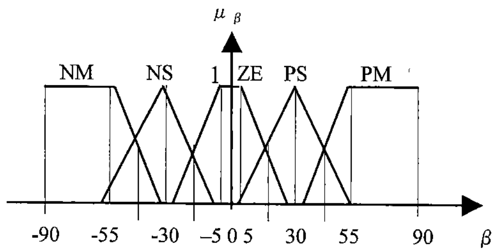
and in Fig.9(b), the spread $S_{a,j+1}$ is updated by

$$\begin{cases} S_{a,j+1} = S_{a,j} + \lambda(C_{b,j} - S_{a,j}), \text{sgn(y - y*)} = \text{sgn}(O_a\text{-}O_b) \\ S_{a,j+1} = S_{a,j} + \lambda(S_{b,j} - S_{a,j}), \text{otherwise} \end{cases} \quad \cdots\cdots(16)$$

Here, δ and λ are learning rates and within [0, 1). y and y* are desired and real output of the training data set. O_a and O_b are the outputs of membership functions μ_a and μ_b. That is to say, the spread point $S_{a,j+1}$ will be moved either towards the center $C_{b,j}$ or the spread point $S_{b,j}$ based on the conditions given in Eq.(16).



(a) Updated membership functions of μ_β



(b) Decided membership functions of μ_β

The dot lines have the same positions with Fig.7.

Fig. 10. Fuzzy membership functions of μ_β.

The adaptive feature of AFCNN with the change of the environment is considered in this section. In the first part of the guideline, except for some initial sampling points, the data fell mainly into the grids of β (t) ='ZE' and the grids of {NS, ZE, PS} for both fuzzy sets β (t) and $\Delta\beta$ (t). Followed by the second part of guideline, the data of fuzzy rules distributed obviously in the edge grids because of the control should be strong.

With this reason, by training the data, AFCNN modifies the fuzzy rules to the system and updates the centers and widths of the membership functions adaptively. Compared to Fig.7, the feature of the membership function for the variable β (t) has been changed shown in Fig.10(a). The widths of edge grids NM and PM become larger adaptively so as to get stronger control results.

Limited by the length of paper, Fig.10 only gives the result of updating for the fuzzy variable β (t). Based on the fuzzy theory and reasonable simplifying, from Fig.10(a), Fig.10(b) is determined as the membership function for the variable β (t), so as to control MV to track the guideline. From Fig.10 we also find that the fuzzy membership functions are not symmetrical because of the mechanical structure of MV and the feature of the guideline.

4.5 Analysis Based on the theory and experiments, AFCNN is proved to be useful in tracking a desired trajectory. Because of its adaptive ability, even if the guideline is a complex curve, as long as it can be tracked, AFCNN can also work well.

Thus the fuzzy rules and sets can be adjusted from the sampled input-output data sets. And if the data are sampled in on-line fashion, fuzzy rules can be modified with the change of real environment in time.

Table 4 and Fig.10 give the distribution of the real fuzzy data sets and the feature of the fuzzy membership function. From them, the following reality can be understood:

(1) Initially, MV may have large tracking error to the guideline, so that the fuzzy control rules should be strong. The first several data of control sequences are distributed in the edge grids of Table 5.
(2) Gradually, the fuzzy rules concentrate to the central grids with the decreasing of tracking errors.
(3) The fuzzy rules become complicated and the membership functions are adjusted greatly when the guideline is a curve than a straight line.
(4) Based on the fuzzy control theory, some control rules are so important that even the numbers of them are small, they should be taken into account (for example, some edge grids and fuzzy rules for β (t) = 'ZE').
(5) It also shows that the segmentation of fuzzy space is not average. Edge grids are larger than central grids. And compared Fig.7 to Fig.10, the centers and widths that segment the fuzzy states can be adjusted.

So AFCNN is not only used as the controller to get better control results (although it has this ability shown in Fig.8), but also for fuzzy rule optimizing and fuzzy feature updating.

5. Conclusions

Automatic control is important in modern society. All of the systems - from home electrical equipments and automobiles, robots, to spacecrafts and nuclear reactors - require automatic control systems to maintain their operation. The stability of the control system and/or the convergence of tracking errors are important. Also the system needed should be adaptable to the changes of the environment and the variations in the actual system dynamics.

Dynamic systems are always represented mathematically, by either differential or difference equations to represent the behavior in continuous or discrete time. These equations provide the changes of variables with the variation of time. There are many techniques for generating control strategies under the condition that the model of the system is known. When the model is unknown, a hybrid control system, which is to design a system with acceptable performance over a very large range of uncertainty greater than can be tolerated only by algorithms for adaptive control systems, is useful and important.

In this paper, the AFCNN system is studied starting from the fuzzy logic and NNs. In fact, one of the main goals of a neuro-fuzzy system is to combine the linguistic representation of fuzzy systems and learning capability of NNs. The AFCNN model uses the learning ability of the neural network and reasoning ability of the fuzzy control for complex, nonlinear and imprecisely defined processes.

For tracking the experimental MV, the problems of real time image data sampling, storage, on-line learning and decision should be solved. These characteristics are essential for an intelligent MV control system. This paper solves the problem of on-line learning and decision. By merging the advantages of the neural network, adaptive (clustering) algorithm and the fuzzy control, AFCNN is presented. It can deal with a large amount of training data by the neural network, from these data producing more reasonable fuzzy rules by the adaptive algorithm, and controlling the object by fuzzy control. It is not the simple combination of the three methods, but merging them into one control system. This research field is still young, but it is getting more important.

By AFCNN, both the structural and linguistic complexity could be reduced by refining the rules. Thus the tracking system is able to track the targets within a certain range of errors. Some conclusions are given. First, it is the validity of AFCNN. Selected from the candidates of the fuzzy rules, they are not destroyed but optimized. Second is its adaptive character. Fuzzy features (the center and width) of each fuzzy membership function can be updated with the change of the environment.

Within this paper, three types of hybrid systems and fuzzy models are introduced and a fuzzy rule producer is given. Also two kinds of adaptively clustering methods are introduced. These are presented at the least level that the general idea of the paper can be obtained. At last, after the introduction of the former control method to MV, AFCNN is given and the experimental result and comparison are gotten. To explain AFCNN clearly, the result is presented by two sections for the fuzzy rule optimizing and fuzzy membership function updating, respectively. Some assumptions and analysis are also considered based on the experiment. At last, in Appendix the original theory of AFCNN is given.

Future work will consider adding the ART neural network as pattern recognizer [20], so that the targets and obstacles can be recognized and some decision can be drawn by the system. Thus AFCNN has the ability to track uncertainly or unknown trajectory of the target or guideline, as long as it is a physically reasonable system. The final object is to adaptively and intelligently control MV, both accurately and quickly.

(Manuscript received October 30, 2002)

References

(1) P. D. Wasserman: Neural Computing: Theory and Practice, Van Nostrand Reinhold (1989)

(2) D. Luo: Adaptive Control, Chinese Railway Press (1993) (in Chinese)

(3) G. Goebel: "An Introduction to Fuzzy Control Systems", http://vectorsite.tripod.com/ttfuzy.html (2000)

(4) M. Hagihara: Neuron, fuzzy, GA, Sangyou Press (1994) (in Japanese)
萩原将文：ニューロ，ファジィ，遺伝的アルゴリズム，産業図書 (1994)

(5) S. Abe: Neural Network and Fuzzy: Theory and Applications, Kindai Scientific Press (1995) (in Japanese)
阿部重夫:ニューラルネットとファージィシステム:理論と応用, 近代科学社 (1995)

(6) A. Gray: "Hybrid Systems FAQ", University of Otago at http://divcom.otago.ac.nz:800/COM/INFOSCI/SMRL/people/andrew/publications/faq/hybrid/hybrid.htm (1997)

(7) K. Watanabe and K. Izumi: "Construction of fuzzy behavior-based control systems for a mobile robot", *Proc. the Third International Symposium on Artificial Life and Robotics*, Vol. 2, pp. 518-523, Oita, Japan (1998-1)

(8) K. Izumi and K. Watanabe: "Fuzzy behavior-based tracking control for an omnidirectional mobile robot", *Proc. the Third International Symposium on Artificial Life and Robotics*, Vol. 2, pp. 524-527, Oita, Japan (1998-1)

(9) H. Wang and C. T. Goh: "Fuzzy logic Kalman filter estimation for 2-wheel steer able vehicles", *Proc. the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 88-93, Kyongju, Korea (1999-10)

(10) X. Wang: Development of Intelligent Control Strategies for a Mobile Vehicle, Ph.D thesis, Division of Materials Science and Production Engineering, Oita University, Japan (1998)

(11) M. Sugisaka and X. Wang: "Artificial brain for a mobile vehicle", *Proc. the Third International Symposium on Artificial Life and Robotics*, Vol. 1, pp. 275-278, Oita, Japan (1998-1)

(12) User's Manual for 98SMC(2)V Two-axes Steering Motor Controller AZI-7201, Interface Corporation (1991) (in Japanese)

(13) T. Takagi and M. Sugeno: "Fuzzy Identification of Systems and Its Applications to Modeling and Control", *IEEE Trans.* Syst., *Man, Cybern.*, Vol. SMC-15, No. 1, pp. 116-132 (1985)

(14) T. Ojala: Neuro-fuzzy Systems in Control. Master of Science Thesis. Department of Electrical Engineering, Tampere University of Technology, Finland (1994)

(15) J. S. Wang and C. S. G. Lee: "Efficient neuro-fuzzy control systems for autonomous underwater vehicle control", Proc. *the 2001 IEEE International conference on Robotics & Automation*, pp. 2986-2991, Seoul, Korea (2001-5)

(16) L.G. Heins and D.R. Tauritz: "Adaptive Resonance Theory (ART): An Introduction", IR-95-35, May/June, 1995, Department of Computer Science, Leiden University,
http://www.wi.leidenuniv.nl/art/#Bibliography

(17) L. Jiao: Neural Network Application and Realization, Xi'an Electronic Science and Technology University Press (1993) (in Chinese)

(18) C. Hennig: "Fixmahal – a software for fixed point cluster analysis by means of the mahalanobis distance",
http://www.math.uni-bamburg.de/home/hennig/ (2002-5)

(19) "LVQ_PAK: The learning vector quantization program package (Version 3.1)", Prepared by the LVQ Programming Team of the Helsinki University of Technology, Laboratory of Computer and Information Science, Finland (1995-7)

(20) S. Hu, Q. Shen, D. Hu, and C. Shi: Neural Network Application Technology, National University of Defence Technology Press (1993) (in Chinese)

Appendix

This is the original theory of AFCNN: Based on sufficient representative training data, reasonable fuzzy rules can be obtained by clustering algorithm.

Followed by the change of working conditions and environment, sampled training data vary correspondingly, and then the fuzzy rules should be adjusted based on them. That is the theory of adaptive fuzzy control system. Since data are trained by NN, it is called AFCNN.

A two-input one output fuzzy control system is given in Fig.A1. The fuzzy rule is to map two input fuzzy vectors A and B to an output fuzzy vector C. Define the dimensions of the fuzzy variables A, B and C are d_a, d_b and d_c, respectively. (A_i, $B_i \rightarrow C_i$) represents one of the fuzzy rules.

Since the value of each fuzzy membership function is within [0, 1], from the viewpoint of geometry, (A_i, $B_i \rightarrow C_i$) can be considered as the map from two points A_i (in a d_a-dimension unit cube I^{d_a}) and B_i (in a d_b-dimension unit cube I^{d_b}), to a point C_i in a d_c-dimension unit cube I^{d_c}.

If we combine A_i, B_i and C_i into one vector (A_i, B_i, C_i), then it is a point in a unit hypercube of $I^{d_a+d_b+d_c}$. This process is AFCNN step (1) in section 3.2.

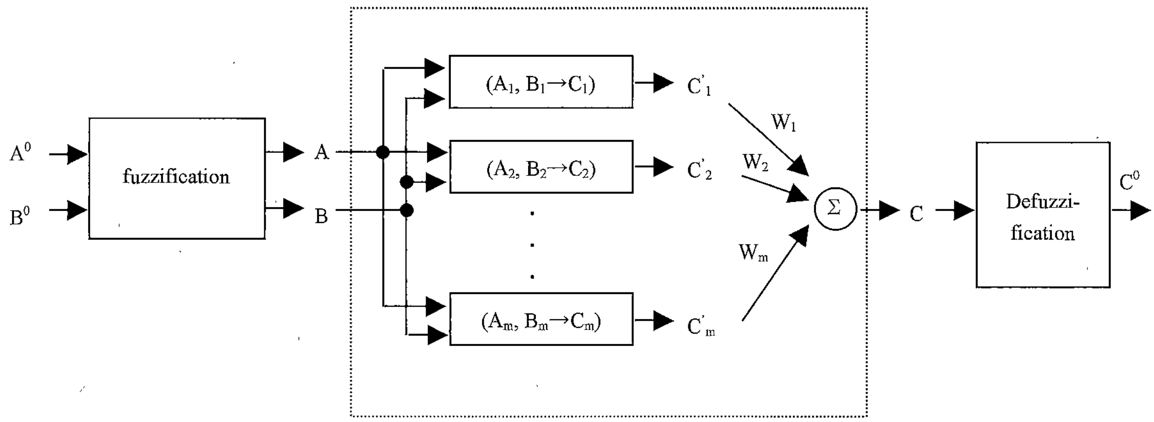


Fig. A1. Two-input one output fuzzy control.

Then the fuzzy rule of $(A_i, B_i \rightarrow C_i)$ is defined by a region in $I^{d_a+d_b+d_c}$, and its center is (A_i, B_i, C_i). If a point A' closes to A_i in I^{d_a} and B' closes to B_i in I^{d_b}, then there is one point C' that will close to C_i in I^{d_c}. In space $I^{d_a+d_b+d_c}$, it is the point (A', B', C') close to (A_i, B_i, C_i). Thus the fuzzy rule $(A_i, B_i \rightarrow C_i)$ can be obtained by define a region in $I^{d_a+d_b+d_c}$ centered by (A_i, B_i, C_i), that is AFCNN step (2) in section 3.2.

After we have obtained sufficient training data of (A_i, B_i, C_i), we can get fuzzy rules by training data in $I^{d_a+d_b+d_c}$. These are step (3) and (4) in section 3.2.

From the above theory and the distribution of the data in the grids, we can also see that: when MV moves smoothly after the initial stage, within each grid, the data have the tendency to distribute around a virtual central point (A_i, B_i, C_i) in the space of $I^{d_a+d_b+d_c}$. Based on this central point and the distribution of data in one grid, the maximum and minimum values for each fuzzy variable can also be updated to make them more reasonable by LVQ2.1 – based on density of distribution to determine the bounds (width) of the membership function. Some results are obtained, but it still needs more deep theoretical analysis, programming and experimental certifying, which is the next step in our project.

Masanori Sugisaka (Member) He received the D. Eng. Degree in Electrical Engineering from Kyushu University in 1975. From 1973 to 1974, he was research associate at Oita University. Since 1988, he has been a professor at Oita University. Now he belongs to the Graduate School of Electrical and Electronic Engineering, Oita University. His current research interests are in the fields of artificial life and robotics, especially artificial brain. He is a member of the Institute of Electrical Engineers of Japan, a member of SICE and a member of IEEE.

Fengzhi Dai (Non-member) was born in Tianjin, China, on February 8, 1971. He received the M.Sci. degree in Department of Automatic Control from Beijing Institute of Technology, China in 1998. From April 1999, he has been a Ph.D student in Division of Materials Science and Production Engineering, Graduate School of Engineering, Oita University. His current research interests are image processing, pattern recognition, neural network and intelligent control.