Proceedings of The
IEEE International Conference
on Industrial Technology, 1996

Design of a Neural-Fuzzy Controller Based on Fuzzy Differential Competitive Learning

Ge Ming
Institute of Industrial Process Control
Hangzhou, 310027, CHINA

Sun YiuXian
Institute of Industrial Process Control
Hangzhou, 310027, CHINA

Abstract— In this paper, a novel neural-fuzzy controller based on fuzzy differential competitive learning is proposed. Since one of the most important parts is the generation of the fuzzy rules in the design of the fuzzy control system, a fast learning algorithm-Fuzzy Differential Competitive Learning(FDCL) for the generation of the rules is applied in the fuzzy control system. The FDCL algorithm adopts a principle of learn according to how well it wins. Unlike the previous competitive learning algorithm such as crisp competitive learning algorithms where only one neuron will win and learn at each competition step every neuron in the neural network based on FDCL algorithm will along with its different distance to the input pattern and learns the pattern accordingly. Compared with the ordinary competitive learning algorithm, the proposed FDCL algorithm has the various distinguishing features. The FDCL algorithm is implanted in the neural network based fuzzy system and the network adopted is fuzzy associative memory system(FAMS) which simulates the knowledge representation and inference process by using fuzzy notation and by association in neural networks. In FAMS the fuzzy rules will be generated by clustering the input-output training data through the FDCL paradigm. By using the FDCL algorithm the neural network can highly refine knowledge and represent the expert experience.

I. Introduction

During the past decade, fuzzy logic control has been widely used in many various fields such as industrial process control. Fuzzy logic control is usually based on a set of fuzzy rules that sum up expert's common sense and experience. But sometimes people maybe find it is difficult to get adequate fuzzy rules, especially when certain complicated dynamic processes are concerned. Now there are many approaches to obtain the fuzzy rules of some specific processes[1][2][3] and the fuzzy logic system designed based on these methods can also been successfully applied, however, these ways are quite problem dependent, i.e., a method may work well for one problem but is not suited for another problem. The other drawback of current fuzzy logic control is that there is no systematic procedure for the design of fuzzy logic control system and thus make it very difficult for people to analyze the properties of the fuzzy logic system. In recent years, the artifical neural network has been widely combined with the fuzzy logic system for it's learning capability and parallel structure. But in these research[4][5] on the neural-based fuzzy logic control system some problems exist: (1) The fuzzy rules identified by the neural networks are hard to understand because these rules are implicitly acquired in the networks; (2) The learning of the networks is time consuming; (3) Some informating concerning the process are mising or not used in the identification of the fuzzy rules.

This paper presents a new kind of clustering methodc Fuzzy-Differential-Competitive-Learning(FDCL) to obtain the fuzzy rules based on fuzzy associative memories(FAM) system. The FAM system provided by Kosko, which integrates neural and fuzzy logic and some learning laws, is used to learn the causal structure of the system[6]. In the designing of the FAM system , it is most important to select some efficient learning law to clustering the fuzzy rules. The clustering methods we proposed here is based on a kind of competitive learning and it's effectiveness is verified through the control of the inverted pendulum system.

In Section II, we propose a four step procedure for generating fuzzy rules from sampled input-output data pairs by the FDCL method. In Section III, the new clustering method is applied to an inverted pendulum control system and this approach is compared with the FAM system using other clustering methods such as the Differential-Competitive-Learning method. Conclusions are given in Section IV.

II. Fuzzy-Differential-Competitive-Learning-Algorithm

The goal of the differential competitive learning is to cluster or categorize the training patterns into some representative groups so that patterns within a cluster are more similar to each other than patterns belonging to different clusters. The differential competitive learning has been suggested as an alternative approach to various sophisticated problems such as the control of nonlinear, time-varying, ill-defined systems. The model is usually associated with a layered neural feedforwad network in which hidden neurons compete according to some sort of distance metric, usually the Euclidean one, to learn the current input pattern $\vec{x}_k$. If the jth neuron wins, it's parametric vector $\vec{m}_j$ is updated additively by some proportion of the difference vector $\vec{x}_k - \vec{m}_j$. The differential competitive learning algorithm is described below and then we will make some modification which lead to the new version of the competitive learning algorithm named Fuzzy-Differential-Competitive-Learning (FDCL) algorithm.

Differential Competitive Learning law (DCL):

$$\dot{m}_j = I_j \Delta S_j(y_j)[x - m_j]$$

0-7803-3104-4

or:

$$m_j(t+1)=m_j(t)+c_t\Delta S_j(y_j(t))[X(t)-m_j(t)] \qquad (1)$$
$$m_i(t+1)=m_i(t) \qquad if \quad i\neq j$$

where c_t is the learning rate and $\Delta S_j(y_j(t))$ denotes the time change of the jth neuron's competitive signal $S_j(y_j(t))$ in the competition field F_Y:

$$\Delta S_j(y_j(t))=S_j(y_j(t+1))-S_j(y_j(t)) \qquad (2)$$

In practice we often use only the sign of the signal difference (2) or sgn(Δy_j), the sign of the activation difference[6]. Kosko has pointed out that the F_Y neuronal activations y_j can be updated with an additive model:

$$(y_j(t+1))=(y_j(t))+\sum_{i=1}^{n}S_i(x_i)m_{ij}(t)+\sum_{i=1}^{m}S_k(y_k)w_{kj} \qquad (3)$$

The fixed competition matrix W defines a symmetric *lateral inhibition* topology within F_Y. In the simplest case w_{ij}=-1 and w_{ji}=1 for distinct i and j.

The differential competitive learning adaptively quantifies the input pattern space R^n and it has been proved that the competitive synaptic vectors m_j can convérge exponentially quickly to pattern-class centroid [6]. The clustering algorithm using the differential learning law is described below:

1.Set the number of competing neurons and initialize the neuron's synaptic vectors: $m_i(0)=x(i), i=1,\ldots,m$.

2. For random sample x(t),find the closest("winning") synaptic vector m_j(t):

$$\|m_j(t)-x(t)\|=\min_i\|m_i(t)-x(t)\| \qquad (4)$$

where $\|x\|$ is the distances between competing neurons, there are many different distance metric can be employed and the most common distance metric is the squared Euclidean norm:

$$\|x\|=\sqrt{x_1^2+\cdots x_n^2} \qquad (5)$$

3. Update the winning neuron's parametric vector using the differential competitive learning law (1).

Although there are many successful applications using the DCL algorithm, the *exclusive* learning mechanism has two drawbacks : one is the neuron underutilizing problem [7][8], and the other is that the information concerning the closeness of input patterns and competing neurons is wasted during the winner-take-all training process because the winner takes all the responsibility for learning the current input pattern in the DCL algorithm. The differential competitive learning law can be rewritten as other form:

$$m_j(t+1)=m_j(t)+c_tI_j\Delta S_j(y_j(t))[X(t)-m_j(t)]$$
$$I_j=\begin{cases}0 & if \quad i\neq j\\ 1 & if \quad i=j\end{cases} \qquad (6)$$

As shown in (6), the indicator function I_j is an crisp(hard) function so that in the competing process only one neuron will win and learn the current training pattern. Obviously, the concept of win in this setting is a crisp one and has a very clear-cut boundary. By considering win as a fuzzy set, every neuron to a certain degree wins, depending on its distance to the current training pattern. Therefore it has to learn according to its win membership during the competition. In this way , we can replace the none-fuzzy indicator function with a fuzzy indicator function, that means, the indicator function is a fuzzy scaling function specifying the sign and magnitude of the difference vector

Below we will derive the fuzzy differential competitive learning via minimization of an objective function. Let us study a collection of n patterns constituting vectors in the p-dimensional space of real numbers, namely $x_1, x_2,\ldots, x_n \in R^p$. We assume that the structure contains c clusters. The Objecting function will be introduced as the following sum:

$$J=\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m\delta_{ik} \qquad (7)$$

subjected to:

$$\sum_{i=1}^{c}u_{ik}=1 \qquad \forall k;$$
$$u_{ik}\in[0,1] \qquad \forall k;i$$
$$0<\sum_{k=1}^{n}u_{ik}<n \qquad \forall k;$$

where :

$u_{ik}\in[0,1]$ denotes the grade of membership of the k-th pattern in the i-th cluster.

The parameter m is used to control the influence of intermediate membership values on the performance index; $1<m<\infty$. The function δ_{ik} is usually defined as the distances between the k-th pattern and the generic structure of the clusters which involves both points and linear varieties:

$$\delta_{ik}=(1-g)D_{ik}^2+g\,d_{ik}^2,$$
$$d_{ik}=\|x_k-m_i\| \quad ,$$
$$D_{ik}^2=\|x_k-m_i\|^2+\sum_{j=1}^{r}(\langle x_k-m_i,s_{ij}\rangle) \qquad (8)$$
$$g\in[0,1]$$

where:

The function D_{ik} involves d_{ik} and adds also a linear variety of dimension r expressed by a scalar product $<\cdot, \cdot>$. This variety goes through the vector m_j and is spanned by the collection of r linearly independent vectors s_{ij}[10]. The role of the parameter g is to keep the balance between these two components(d_{ik} and D_{ik}).

By applying the gradient descent method to the objective function J, a fuzzy differential competitive learning law can be obtained:

$$m_j(t+1) = m_j(t) + c_t I_j \Delta S_j(y_j(t))[X(t) - m_j(t)] \quad (9)$$

$$I_j = (u_{ik})^m$$

$$u_{ik} = \frac{1}{\left\{ \sum_{j=1}^{c} \left(\delta_{ik} / \delta_{jk} \right) \right\}^{1/(m-1)}} \quad (10)$$

Since the derivation of these formulae is the same as that of the fuzzy clustering algorithm that has already been studied in [10] so the procedure of this derivation is not discussed in detail. From the derived algorithm , it can be seen that every neuron is responsible for learning the current training pattern rather than only a winning neuron has the responsibility. It is already mentioned in the front of this section that normal differential competitive learning has two disadvantages and the disadvantages maybe lead to the failure of the learning, however, the fuzzy differential competitive learning will reduce the probability of the learning failure and its effectiveness will be demonstrated in the section Ⅲ. From the equation (9) and (10) we can see that the membership u_{ik} is changed with the distance between the neuron and the pattern, that is, the closer the neuron to the pattern, the larger its win membership u_{ik} is and the more it has to learn. However, the far-away neurons has also the chance to update its synaptic vector and can be moved to some pattern regions. Therefore the problem of the old differential competitive learning that it waste the information concerning the closeness of input patterns and competing neurons is solved. In the section Ⅲ, we will integrate this new clustering algorithm with the FAM system and develop a fuzzy control system.

Although it has been proved that the synaptic vectors converge to decision-class centroids that correspond to local maxima of the sampled but unknown probability density function $p(x)$ through the differential competitive learning [6], the fuzzy differential competitive learning is not yet proved and will be studied. In the following discussion we will only study the centroid theorem because it is very important for the fuzzy clustering process and the others are discussed similarly by Ksoko[6].

Suppose that the decision classes D_1, D_2, ..., D_k partition R^n into k classes:

$$R^n = D_1 \cup D_2 \ldots \cup D_k \quad (11)$$

$$D_1 \cap D_2 \neq 0 \quad \text{if } i \neq j \quad (12)$$

Centroid defines the deterministic "center of mass" of pattern class D_j [6]:

$$\underline{x}_j = \frac{\int_{D_j} \Delta S_j(y_j) X p(x) dx}{\int_{D_j} p(x) dx} \quad (13)$$

The centroid theorem will be first proved that if a fuzzy differential competitive learning system converges, it converges to the centroid of the sampled decision class.

Centroid theorem:

$$\text{Prob } (m_j = \underline{x}_j) = 1 \text{ at equilibrium} \quad (14)$$

where $\underline{x}_j$ is the centorid of the sampled decision class.

proof: Suppose the jth neuron in F_Y wins the activation competition during the training interval and the jth synaptic vector m_j codes for decision class D_j. The stochastic fuzzy differential competitive learning equations is adopted in the proof for In practice we actually use the random process for studying. The stochastic fuzzy differential competitive learning law (9) can be rewritten as :

$$\dot{m}_j = I_j \Delta S_j(y_j)[x - m_j] + n_j \quad (15)$$

where $\{n_j\}$ represents a zero-mean Gaussian white-noise random process. I_j is defined in (9).

Suppose the synaptic vector m_j has reached equilibrium:

$$\dot{m}_j = 0 \quad (16)$$

which holds probability one. Take expectations of both sides of (15), use the zero-mean property of the noise process, eliminate the synaptic vector $\dot{m}_j$ with the stochastic fuzzy differential competitive leaning (14) and expand to give :

$$\begin{aligned}
&\because E[\dot{m}_j] \\
&= \int_{R^n} I_j \Delta S_j(y_j)(X - m_j) p(x) dx + E[n_j] \\
&= \int_{D_j} \Delta S_j(y_j)(X - m_j) p(x) dx \\
&= \int_{D_j} \Delta S_j(y_j) X p(x) dx - \int_{D_j} m_j p(x) dx \\
&= \int_{D_j} \Delta S_j(y_j) X p(x) dx - m_j \int_{D_j} p(x) dx \\
&= 0
\end{aligned}$$

$$\therefore m_j = \frac{\int_{D_j} \Delta S_j(y_j) X p(x) dx}{\int_{D_j} p(x) dx}$$

Thus the theorem is proved.

Ⅲ. Application to the Inverted-Pendulum System

In this section, we will give an simulation of the control of the Inverted Pendulum System with the Fuzzy-Differential-Competitive-Learning-Algorithm(FDCL) and the normal Differential-Competitive-Learning-Algorithm (DCL).

Controlling of the Inverted Pendulum System is a hard task because the pendulum system is a non-linear system for which there are no traditional control system design methods. Ksoko has proposed the fuzzy control strategy for the pendulum system with the DCL algorithm. Now we use the FDCL algorithm to control the same problem and compared it with the DCL algorithm.

The Inverted Pendulum System is composed of a rigid pole and a cart on which the pole is hinged. The cart moves on the rail tracks to its right or left, depending on the force exerted on the cart. The control aim is to balance the pole starting from nonzero conditions by supplying appropriate force the cart. The dynamics of the Inverted Pendulum System are characterized by four state variables: θ (angle of the pole with respect to vertical axis), θ '(angular velocity of the pole), z (position of the cart on the track), and z'(velocity of the cart). The relationship between these variables is determined by two second-order differential equations described by [9].

$$\ddot{\theta}=\frac{g*\sin\theta+\cos\theta*\left(\dfrac{-f-m*l*\ddot{\theta}*\sin\theta}{m_c+m}\right)}{l*\left(4/3-\dfrac{m*\cos^2\theta}{m_c+m}\right)} \tag{16}$$

$$\ddot{z}=\frac{F+m*l*\left(\dot{\theta}^2*\sin\theta-\ddot{\theta}*\cos\theta\right)}{m_c+m}$$

We use the equations to generate the desired Input-Output data pairs(θ , θ '; F)(F is the force that balance of the pendulum system). Because the detailed process of controlling the pendulum system and the constructing of the FAM(Fuzzy Associative Memory) system is similar to the previous methods proposed by Ksoko except for the implanted learning algorithms, thus we only give the results of simulation of these two learning algorithms(FDCL and DCL). Figure1 shows the results of the two different control methods.

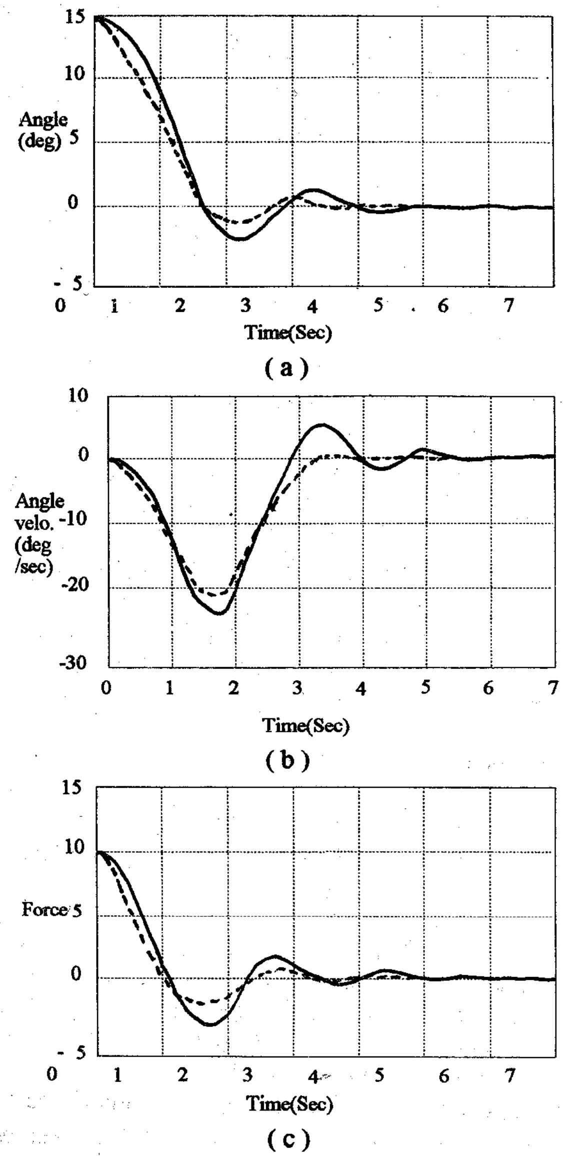

Fig. 1. (a) Pole angle, (b) pole angular velocity, (c) balance force. (Solid, dashed curves with regard to different learning methods: DCL, FDCL, respectively.)

IV.CONCLUSIONS

In this paper, we have proposed a Fuzzy-Differential-Competitive-Learning-Algorithm(FDCL) and implanted it in the FAM system for constructing a Neural-Fuzzy controller. By comparing of the results of the two simulations with two learning algorithms we can find that there are some improvements by using the FDCL algorithm.

V. REFERENCE

1. E.M. Scharf and N.J. Mandic, "The application of a fuzzy controller to the control of a multi-degree-freedom robot arm," *Industrial Application of Fuzzy Control, M.Sugeno, Ed. Amsterdam: North Holland,* 1985, pp. 41-62.

2. S. Shao, "Fuzzy self-organizing controller and its application for dynamic processes," *Fuzzy Sets Syst.,* Vol.26, pp. 151-164, 1988.

3. R. Tanscheit and E.M. Scharf, "experiments with the use of a rule-based self-organizing controller for robetics applications," *Fuzzy Sets Syst.,* Vol.26, pp. 195-214, 1988.

4. A.G. Barto, R.S. Sutton, and C.W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst., Man, Cybern.*, Vol. SMC-13, no.5. pp.834-847, 1983.

5. F.C. Chen, "Back-propagation neural network for nonlinear self-tuning adaptive control," *Proc. IEEE Intelligent Machine,* pp. 274-279, 1989.

6. B. Kosko, *Neural Networks and Fuzzy Systems.* Englewood Cliffs, NJ: Prentice-Hall, 1992.

7. Ahalt, S.C., Krishnamurthy, A.K.. Chen, P..& Melton, D.E., "Competitive Learning algorithms for vector quantization, " *Neural Networks*, 3, pp.277-290, 1990.

8. Grossberg, S., "Adaptive pattern classification and universal recoding: 1. Parallel development and coding of neural feature detectors," *Biological Cybernetics*, vol. 23, pp121-134.

9. A.G.Barto, R.S.Sutton, and C.W.Anderson, "Neronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst. Man.Cybern*, vol. SMC-13, no. 5, pp.834-846, 1983.

10. Y.Yoshinari, W.pedrycz, and K. Hirota, "Construction of Fuzzy models through clustering techniques," *Fuzzy Sets and Systems*, Vol.54, pp. 157-165, 1993.