

A Variable Structure Compensation Fuzzy Neural Network and Identification

Jing LIU*, Yalu CAO*, Ping WANG*, Li PENG**, RenChun Lian***, Xixin CAO*

* School of Software and Microelectronics, Peking University, China

** School of IoT Engineering, Jiangnan University, China

***Inspeed Communication Co.,Ltd, China

liujing@ss.pku.edu.cn, pengli@jiangnan.edu.cn, cxx@ss.pku.edu.cn

Abstract— The compensation fuzzy neural network (CFNN) with fast learning algorithm and compensation fuzzy inference is proposed in this paper. And a new variable structure compensation fuzzy neural network (VS-CFNN) is proposed on the basic of CFNN. The fuzzy rules of the nodes are changed with the input nodes. The triangular function and parameter choosing methods have been given. At the mean time, the parameters have also been adjusted through BP algorithm. Simulation results verify that the VS-CFNN has higher identification precision and faster identification speed than the ordinary FNN in identification. The proposed method has also been implemented to identify the Krasnosel'skii's hysteron hysteresis model parameters of an electrical valve actuator installed on a pneumatic system. Three types of sensors (actuator position, air pressure and mass airflow rate) are used to investigate the hysteresis model parameter identification using the proposed VS-CFNN method. The experimental results have demonstrated the effectiveness of the proposed method.

Keywords— CFNN, variable structure, learning algorithms, identification

I. INTRODUCTION

Fuzzy neural network(FNN) as a cross subject made up with fuzzy set theory, neural computation, digital signal processing, random process, and pattern recognition is still a hot research direction(e.g. [1]-[4]). The main characteristic of using FNN is to modulate the grades of membership function and logic rules in fuzzy logical reasoning system. Then a forward propagation structure of NN will be constructed by using the format of fuzzy logic reasoning. That is to bring experience into the choice of NN initial structure and weights, thereby it can exert their advantages and realize the compensation each other.

But the normal FNN has difficulty in designing itself (e.g. [5], [8]). For example, it is hard that how to choose fuzzy membership function, how to choose

fuzzy rule layer node number, how to choose fuzzy logic reasoning and optimal fuzzy operation, and how to choose opposite fuzzy function. To overcome these difficulties, this paper provides a CFNN which has fast study algorithms, and can give compensation with fuzzy logic. Based on it, a new variable structure of CFNN is proposed, which its node number of the fuzzy rules is changed along with the input, and triangle membership function is adopted. At the same time, the method to determine the parameters of fuzzy rules is also given. At the end, the last layer parameters are amended by using BP algorithms.

II. VARIABLE STRUCTURE CFNN

A. Network structure

CFNN is a mix system combined with compensation fuzzy logic and NN, and CFNN is made up with fuzzy neurons which face to control and decision. These neurons are defined to do fuzzy operation, include fuzzy logic, compensation and fuzzy opposite operation. Because the compensation fuzzy neurons are introduced into CFNN, the network can be trained from the fuzzy rules. That will make network fault-tolerant capability better, and bring system more stabilization. At the same time, in normal FNN, static and local optimal operation often occurs in fuzzy operation, such as minimal operation and maximal operation. But in CFNN, fuzzy operation adopts dynamic and global optimal operation. And the fuzzy operation optimizes NN learning algorithms, gives compensation to fuzzy operation in dynamic. For the more, make the network more optimal. Network not only adjusts input and output fuzzy membership function properly, but also optimizes fuzzy logic rules in dynamic by logic arithmetic compensation, and the network parameters have clear physical meaning. We can initialize heuristic algorithms to expedite training speed.

A CFNN has four layers such as input layer, fuzzy layer, compensation fuzzy logic layer and opposite fuzzy layer. The connection between layer and layer is depending on s language variables in fuzzy logic system, compensation fuzzy logic method and opposite fuzzy function. The structure is as Figure 1.

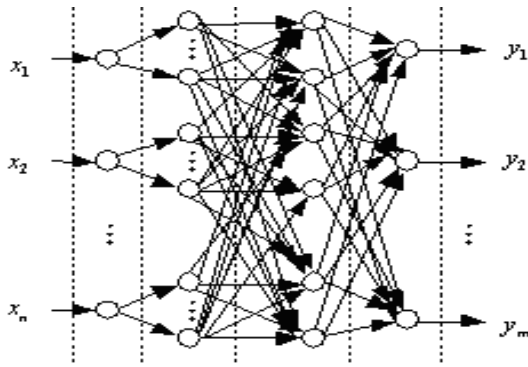


Figure 1. CFNN structure

In Figure 1, every node in the first layer connects with input vector directly. Every node in the second layer deposes one language variable value, whose action is to calculate the membership function. Every input vector belongs to every language variable value in fuzzy set. Every node in the third layer deposes a fuzzy rule, whose action is matching fuzzy rule and calculate every rule's fitness, then to make compensation operation. The fourth layer is opposite fuzzy layer.

B. Parameters determination

1) Fixing membership function parameters

Input space x 's base sets is $[x_i^-, x_i^+]$. Output space y 's base sets is $[y_i^-, y_i^+]$. Then the input space x 's fuzzy division can be carried out as the follow rules, x is subdivided in the sub-space where y changes acutely as along with x changes, x is rough separation in the space where y changes repose fully as along with x . Concretely, the condition that y changes along with x can be analyzed. Using the position of an extremum and the number of point of inflection to make sure the change acuity degree when x changes in some sub-space. The sub-space where position of an extremum and the number of point of inflection is more, the fuzzy division to x is more, and the reverse is true. Firstly, to make sure approximately position of an extremum and the number of point of inflection N and the corresponding input vector value of x_i^k (here i means $i=1,2,\dots,n$, k means fuzzy break point $k=1,2,\dots,N+2$ according to extremum and point of inflection) in the x and y base sets according to the stylebook. Make the input vector x_i^k be every input variable's fuzzy break point, and then the number of the fuzzy break point of every input variable is $N+2$.

Because fuzzy break point may not be bisecting point in input base sets, using Gausse and rhombus function to be membership function. It is very rigor to choose initial value toward CFNN, otherwise the net training will trap in confusion condition, and can not reach the request of error precision, so the

trigonometric function is chosen to be membership function.

$$\mu_i^j = \begin{cases} (x_i - a_{ij}) / (c_{ij} - a_{ij}) & a_{ij} \leq x_i \leq c_{ij} \\ (b_{ij} - x_i) / (b_{ij} - c_{ij}) & c_{ij} \leq x_i \leq b_{ij} \end{cases}$$

$$i = 1, 2, \dots, n \quad j = 1, 2, \dots, m_i$$

the choice of a_{ij} , b_{ij} , c_{ij} should be subject to

$\sum_{j=1}^{m_i} \mu_i^j = 1$. The parameters of the membership once been sure and will not be amended.

2) Fixing the number of rules

Supposing in the second layer, the fuzzy every break point input variable is according to and the number is m_i ($i=1, 2, \dots, n$) well then the number of the rule of traditional CFNN model, namely the number of the third layer's node, is $M = m_1 \times m_2 \times \dots \times m_n$. Obviously, if the input variable is more and every input variable's fuzzy break points are more, the M is large number, and it will prolong the calculation rapid of CFNN model. This shows that if M value can be reduced, the calculation rapid of CFNN model will be improved heavily.

By analyzing, in the approach of every sample calculation, there are at most two variables not zero in the fuzzy language variable's membership function μ_i^j according to CFNN model's every input variable. The rule layer, namely the third layer's output variables have at most 2^i ($i=1, 2, \dots, n$) not zero, which means that there are at most 2^i ($i=1, 2, \dots, n$) rules useful. When the CFNN model's output is calculated, the nodes which are not zero are kept down, the others are deleted or don't join the calculation. To calculate the these nodes' position, firstly make a assumption that in the approach of every sample calculation, the net input is x_i ($i=1,2,\dots,n$), and the second layer's output is $f_{ij}^{(2)}$ ($i=1,2,\dots,n \quad j=1,2,\dots,m_i$). Then if $f_{ij}^{(2)} = 0$, with it as the third layer's corresponding node's output value of the input is zero, and now the node variable needn't be calculated, namely the node should be deleted.

If $f_{ij}^{(2)} \neq 0$, with it as the input the third layer's corresponding node's output value is not zero, and now the node variable should be calculated. These nodes' position size can be expressed by the follow formula.

$$p^{(3)} = \left[\prod_{i=2}^n m_i \right] (p_{1j}^{(2)} - 1) + \left[\prod_{i=3}^n m_i \right] (p_{2j}^{(2)} - 1) + \dots + m_n (p_{(n-1)j}^{(2)} - 1) + p_{nj}^{(2)} \quad (1)$$

There into

$p^{(3)}$ —the third layer's node's position size where the output variable is not zero. m_i —the first layer's fuzzy

break point, $p_{ij}^{(2)}$ —the second layer's node's position

where the output variable is not zero (by the membership function choice rule, there are at most two membership function values of the first layer's every node, then $i=1, 2, \dots, n, j=1, 2$).

After the net structure is confirmed according to the above method, in the approach of every sample calculation, the total numbers of rules is

$M'=2^i$ ($i=1, 2, \dots, n$) at most. So in this paper it is

much less than the traditional CFNN's third layer's node, thereby the computational complexity is reduced, and the rapid of calculation is improved.

3) Confirming last layer's parameters

Essentially CFNN is also a kind of multi-layers feed forward network, so can follow the network of BP to design and adjust the parameter's learning method with the method of error back propagation.

Choose the error cost function.

$$E_i = \frac{1}{2} (y_{di} - y_i)^2 \quad i = 1, 2, \dots, r \quad (2)$$

Thereinto,

y_{di} — CFNN expectation output. y_i — CFNN actual output. E_i — error function. The next $\partial E_i / \partial w_{ij}$ will be calculated by error back propagation. Then using the first order gradient optimization algorithms to adjust w_{ij} .

$$\partial E_i / \partial w_{ij} = -(y_{di} - y_i) \sigma_j \quad (3)$$

After the first order gradient is worked out, the learning method adjusted by parameter is given at last.

$$w_{ij}(k+1) = w_{ij}(k) - \beta \frac{\partial E_i}{\partial w_{ij}} \quad (4)$$

$i = 1, 2, \dots, r \quad j = 1, 2, \dots, m$ in the formula, β —

rate of learning ($0 < \beta < 1$).

4) Learning algorithms

The m if-then rules of compensation fuzzy logic system which has n outputs and one input, which can be expressed as the follow,

$FR^{(k)} : \text{if } x_1 \text{ is } A_1^k \text{ and } \dots \text{and } x_n \text{ is } A_n^k$

then y is B^k

($i = 1, 2, \dots, n \quad k = 1, 2, \dots, m$) (5)

There into, A_i^k is a fuzzy set in the base sets U.

B^k is a fuzzy set in base sets V. x_i and y are language variables.

The fuzzy set B' can be produced in output base sets V according to the Kth fuzzy rule if fuzzy subset

A' is inputted in base sets U. Fuzzy logic adopt algebra product ($\cdot$) operation. The fuzzy set B' derived by fuzzy logic rule in base sets V is

$$\mu_B^k(y) = \mu_{A_1^k \times \dots \times A_n^k \rightarrow B^k}(x, y) \cdot \mu_A^k(x)$$

Fuzzy implication adopt product operation $R_p = A \rightarrow B$, Namely $\mu_{A \rightarrow B}(x, y) = \mu_A(x) \mu_B(y)$.

Define compensation operation,

$$\mu_{A_1^k \times \dots \times A_n^k}(x) = [\prod_{i=1}^n \mu_{A_i^k}(x)]^{1-\gamma+\gamma/n}$$

Thereinto n is input vector's dimension, γ is compensation degree, $\gamma \in [0, 1]$.

$$\mu_B^k(y) = \mu_A^k(x) \mu_B^k(y) [\prod_{i=1}^n \mu_{A_i^k}(x)]^{1-\gamma+\gamma/n}$$

Adopt single value fuzzification $\mu_A^k(x) = 1$. Then

$$\mu_B^k(c_{ij}) = [\prod_{i=1}^n \mu_{A_i^k}(c_{ij})]^{1-\gamma+\gamma/n} = z^k.$$

Thereout opposite fuzzification function can be defined as the follow

$$f(x) = \sum_{k=1}^m c_{ij} w^k z^k / \sum_{k=1}^m w^k z^k \quad (6)$$

So, this system is made up with triangle membership function, compensation fuzzy logic and modified barycenter opposite fuzzification machine.

The aim function is,

$$E = \frac{1}{2} \sum_{i=1}^n (f(x_i) - y)^2 \quad (7)$$

Adopt the gradient descent which adjust step in dynamic to train system's input and output membership function's parameter and compensation operation's compensation degree. The corresponding formula is as the follow,

$$\begin{aligned} a_j^k(t+1) &= a_j^k(t) - \eta(t) \frac{\partial E}{\partial a_j^k(t)} \\ b_j^k(t+1) &= b_j^k(t) - \eta(t) \frac{\partial E}{\partial b_j^k(t)} \\ c_j^k(t+1) &= c_j^k(t) - \eta(t) \frac{\partial E}{\partial c_j^k(t)} \end{aligned} \quad (8)$$

Because of compensation $\gamma \in [0, 1]$, make a suppose as the follow might as well,

$$\gamma_k = \frac{d_k^2}{d_k^2 + e_k^2} \quad k = 1, 2, \dots, m \quad (9)$$

$$\frac{\partial E}{\partial d_k} = \frac{\partial E}{\partial \gamma_k} \frac{\partial \gamma_k}{\partial d_k} = \frac{2d_k e_k^2}{(d_k^2 + e_k^2)^2} \frac{\partial E}{\partial \gamma_k} \quad (10)$$

So (t is the number of training step)

$$\begin{aligned} d_k(t+1) &= d_k(t) - \eta(t) \frac{\partial E}{\partial d_k(t)} \\ &= d_k(t) - \eta(t) \frac{2d_k e_k^2}{(d_k^2 + e_k^2)^2} \frac{\partial E}{\partial \gamma_k} \end{aligned} \quad (11)$$

$$\begin{aligned} e_k(t+1) &= e_k(t) - \eta(t) \frac{\partial E}{\partial e_k(t)} \\ &= e_k(t) + \eta(t) \frac{2d_k^2 e_k}{(d_k^2 + e_k^2)^2} \frac{\partial E}{\partial \gamma_k} \end{aligned} \quad (12)$$

$$\begin{aligned} \gamma_k(t+1) &= \frac{d_k^2(t+1)}{d_k^2(t+1) + e_k^2(t+1)} \\ k &= 1, 2, \dots, m \end{aligned} \quad (13)$$

III. SIMULATION EXPERIMENT

Simulation object is

$$f(u(k)) = 0.7\sin(\pi u) + 0.3\sin(\pi u) + 0.1\sin(5\pi u) \quad u \in [-1, 1] \quad (14)$$

Identify and research traditional FNN and VS-FNN provided in this paper. the Fig. 2 is the simulation curve in MATLAB.

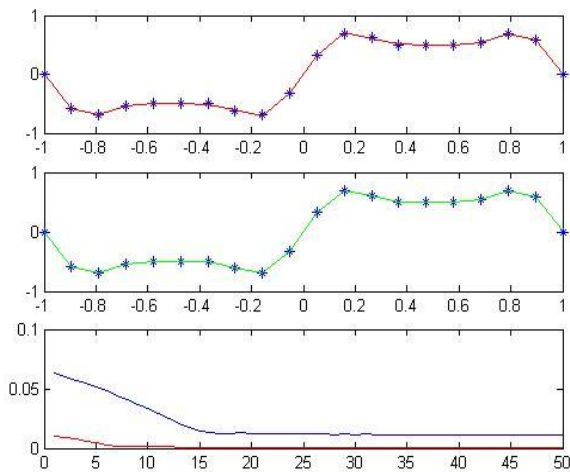


Figure 2. Comparison diagram of simulation results

Here the first diagram is the comparison between actual output and traditional output (* means actual output, red line means traditional method output), the second diagram is the comparison between actual output and modified method output (* means actual output, green line means modified method output), the third diagram is the comparison between traditional method training process error and modified method

training process error (blue line means traditional method error curve, red line means modified method error curve).

From the Figure 2, we can see that the effect of identification of traditional FNN and VS-FNN is obvious, but two methods' identification precision and rapid is different. After the same 50 training steps, the traditional method's training error is 0.010672, but the training error of the method provided in this paper is 0.00031743. With the request that the error precision is 0.001, the identification rapid of traditional method is not fast as the method provided in this paper obviously. So the advantage of VS-FNN method provided in this paper is much obvious.

The VS-FNN method has been implemented to identify the Krasnosel'skii's hysteron hysteresis model (e.g. [9], [10]) parameters of an electrical valve actuator installed on a pneumatic system. Three types of sensors (actuator position, air pressure and mass airflow rate) are used to investigate the hysteresis model parameter identification. The identified parameter values were averaged after the ten runs and are shown in Table 1.

TABLE I. PARAMETER VALUES IDENTIFIED BY GAASS USING VARIOUS SENSOR MEASUREMENTS

Sensor	τ	ζ	ω_n	m_b	$c_{h,l}$	$c_{h,r}$
position	N/A	N/A	N/A	1.09	0.48	1.02
pressure	1.79	N/A	N/A	1.11	0.59	1.15
flow	N/A	0.51	0.22	1.10	0.55	1.09
	$m_{h,l}$	$m_{h,r}$	$m_{h,b}$	$c_{h,l}$	$c_{h,r}$	$c_{h,t}$
position	1.01	1.08	0.08	0.12	-0.28	0.92
pressure	1.08	1.11	0.10	0.10	0.11	1.07
flow	1.01	1.09	0.10	0.13	-0.32	1.04

As seen from Figure 3, the hystereses identified using different sensor measurements have matched well with the predefined hysteresis.

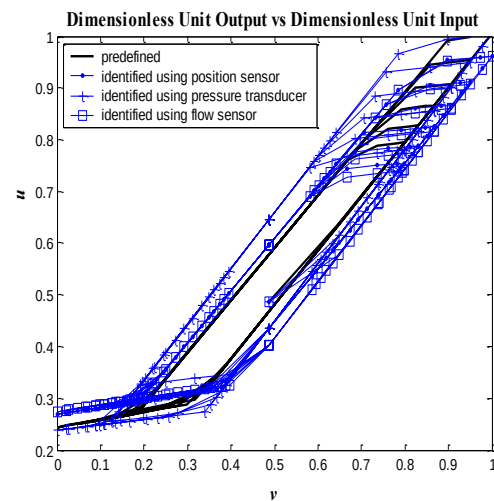


Figure 3. Identification of the hysteresis

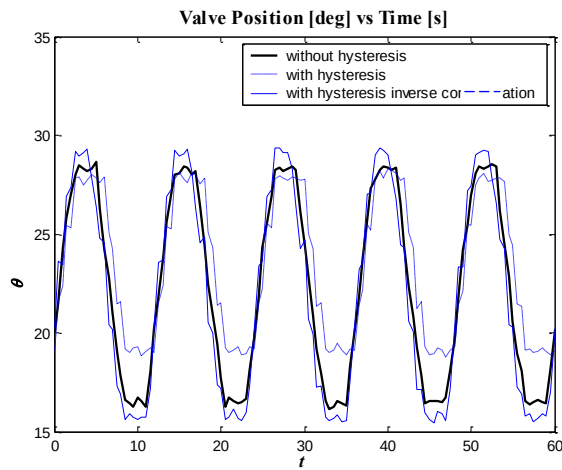


Figure 4. Actuator position control results

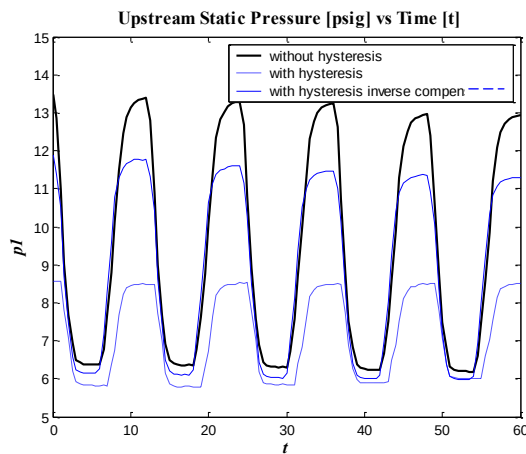


Figure 5. Pressure control results

The identification results plotted in Figure 4-5 have revealed the effectiveness of this joint method of hysteresis model parameter identification and hysteresis inverse compensation when the actuator is controlled for a duration of 60 seconds and a sampling time of 0.5 second. The compensation results have shown that $\varepsilon\% = 3.80$ if the position sensor is used, $\varepsilon\% = 13.27$ if the pressure transducer is used, and $\varepsilon\% = 38.87$ if the thermal flow sensor is used.

The actuator position sensor measurement has led to the most accurate actuator hysteresis parameter identification, which is easily understandable since the sensor response is fast and the pipeline airflow dynamics virtually does not affect the sensor output. The thermal flow sensor contains slow and complicated dynamics, and its output is affected by the pipeline airflow dynamics. It is not surprising to see the actuator hysteresis parameter identification became less accurate when the thermal flow sensor measurements were used. Compared to the thermal flow sensor, the pressure transducer is faster and has led to better actuator hysteresis parameter identification and compensation results, but not as good as those obtained using the position sensor measurements.

The hysteresis inverse compensation based on the model using the identified hysteresis parameters has effectively softened the effect of actuator hysteresis.

IV. CONCLUSION

The author introduces a kind of speediness learning algorithm and CFNN which can do compensation fuzzy logic. And in this base provides a kind of new VS-FNN, and make triangle membership function a use into the network and give out a method which can confirm fuzzy rule parameter. Use the error back propagation to amend the last layer parameter. The simulation experiment shows that the VS-FNN method provided in this paper is effective and this method not only predigests the number of the net fuzzy rule nodes but also improves network identification precision and training rapid. The method has a wide future in actual application.

REFERENCES

- [1] Xianhua Zeng, Xiaofeng Lai: Compensation fuzzy neural network's modification and instance analyse. Sichuan Normal Collage transaction (Natural science). Vol.23No3, (2002)249-252
- [2] Enqin Zhang, Songjiao Shi, Weihua Gao: The research and development of fuzzy control system recent years [J]. Control theory and application. Vol.18No1, (2001)7-11
- [3] Zhihua Zhang, Nanning Zheng, Zhengxuan Zhao: Based on fuzzy object neural network's nonlinearity dynamic system identifier [J]. Control theory and application., Vol.17No4, (2000)565-568
- [4] Hongwei Wang, Guangfu Ma: Based on fuzzy rule nonlinearity dynamic system modeling method. Control theory and application[J]. Vol.17No3, (2000)419-422
- [5] Fang J H, Chen H C: Fuzzy Modelling and The Prediction of Porosity and Permeability from the Compositional and Textural Attributes of Sandstone. Journal of Petroleum Geology. Vol.20No2, (1997) 185
- [6] Huanyao Qiu, Zhongyuan Mao: Variable structure compensation fuzzy neural network control and the learning rule research. Southern China University of Technology transaction (Natural science). Vol.27No3, (1999) 88-93
- [7] Cho S.B. Neural-network classifiers for recognizing totally unconstrained handwritten numerals[J]. IEEE Transactions on Neural Networks, 1997, Vol.8, No.1, pp.43-53
- [8] Purushothaman G. and Karayiannis N. B.. Feed-forward neural architectures for membership estimation and fuzzy classification[J]. International Journal of Smart Engineering System Design, 1998, Vol.1, No.1, pp.163-185
- [9] L.O. Chua and S.C. Bass, A generalized hysteresis model, IEEE Trans. Circuit Theory, CT-19 (1972) 36-48.
- [10] G. Tao and P.V. Kokotović, Adaptive control of systems with actuator and sensor nonlinearities, John Wiley & Sons, Inc., New York, NY, 1996.