J. Marine. Sci. Appl. (2008) 7:157-161
DOI: 10.1007/s11804-008-7081-1

# Advanced FNN control of mini underwater vehicles

**XU Yu-ru (徐玉如), GUO Bing-jie* (郭冰洁), LI Yue-ming (李岳明)**

*College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China*

**Abstract:** Fuzzy neural networks (FNN) based on Gaussian membership functions can effectively control the motion of underwater vehicles. However, their operating processes and training algorithms are complicated, placing great demands on embedded hardware. This paper presents an advanced FNN with an S membership function matching the motion characteristics of mini underwater vehicles with wings. A learning algorithm was then developed. Simulation results showed that the modified FNN is a simpler algorithm with faster calculations and improves responsiveness, compared with a Gaussian membership function-based FNN. It is applicable for mini underwater vehicles that don't need accurate positioning but must have good maneuverability.

**Keywords:** mini underwater vehicle; advanced fuzzy neural network; S membership function
**CLC number:** TP24 **Document code:** A **Article ID:** 1671-9433(2008)03-0157-05

## 1 Introduction

Autonomous underwater vehicles (AUVs) work in unknown ocean environment. Owing to the time-varying of motion and the complexity and uncertainty of environment, it is difficult to construct the dynamic model for underwater vehicles. Furthermore, as the dynamic system of autonomous underwater vehicle is nonlinear and time-varying, it is hard to estimate or measure the hydrodynamic parameters. Therefore, many factors should be taken into account during the design of the control system of an AUV. At the same time, it is very important to strengthen the AUV's autonomy and adaptability to improve its performance. From this point of view, the control system should have the capability of self-adapting and self-learning[1].

FNN combines basic fuzzy logic control cell and network structure of neural network to form a fuzzy neural network that can teach itself. Neural network is used to construct the fuzzy controller. The self-learning method of the neural network is used to realize the self-learning and self-adapting of the fuzzy control.

The traditional FNN takes Gauss function as the membership function. Because Gauss function has the property of radial symmetry and its derivative can be got easily. So if neural network takes it as the membership function, it will have excellent ability of approximating and high convergence speed. However, Gauss function is complicated. In practice, complication of algorithm will greatly affect the incorporated property of FNN[2].

In this paper, we provided the advanced FNN with S membership function-based, which is a compromise between incorporated property and complication of algorithm. The training algorithm of neural network is also changed accordingly.

## 2 Structure of advanced FNN

The advanced FNN consists of 4 layers. As depicted in Fig.1, FNN consists of input linguistic nodes layer, input term nodes layer, rule nodes layer and output linguistic nodes layer[2-5].

A typical rule has the following form:

$$R_l \text{ if } x_1 = A_1^l \text{ and } x_2 = A_2^l, \text{ then } y = \theta_l$$

where $x_1$, $x_2$ and $\theta_l$ are the $l$th fuzzy parameters which are associated with $e$, $\dot{e}$ and y. $e$ and $\dot{e}$ are input variables of FNN. They represent error and error change. $A_1^l$ and $A_2^l$ are values of $x_1$ and $x_2$, the domain of which is $\{-6,-5,\cdots,0,1,\cdots,5,6\}$. $y$ is the output of this system, $l$=0,1,2,$\cdots$,12. The number of rules is 169.

**Received date:** 2007-10-16.
**Foundation item:** Supported by the Fundamental Research Foundation of Harbin Engineering University Foundation under Grant No. HEUFT08001.
***Corresponding author Email:** guobingjie117@163.com

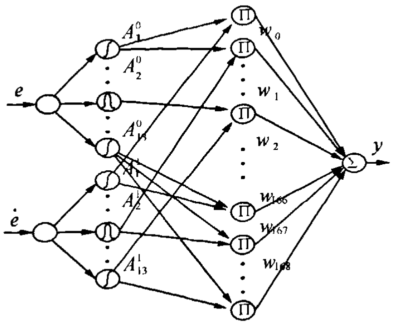


Fig.1 Neural network structure of FNN

The validity $\alpha_l$ of each rule antecedent is evaluated by finding the intersection "AND" between premises in the antecedent. This is accomplished by taking the minimum membership value $\mu$ between the evaluated membership functions for the input partition referenced in each premise:

$$\alpha_l = A_1^l(x_1) \wedge A_2^l(x_2). \quad (1)$$

The final output of FNN is

$$y = \sum_{l=0}^{168} \theta_l \alpha_l w_l, \quad (2)$$

where $w_l$ is the weight of neural network, and the membership function is S function. $x_i^k$ represents $i$th input of the $k$th layer, $\text{net}_j^k$ represents the input of $j$th node in the $k$th layer, $y_j^k$ represents the output of $j$th node in the $k$th layer. It can be known that $y_j^k = x_j^{k+1}$, so the operation in every layer of FNN can be got as follows:

1) The first layer: input linguistic nodes layer.

There are two nodes in this layer. They represent $e$ and $\dot{e}$. Each of them sends the input to the nodes of the next layer directly:

$$\text{net}_i^{(1)} = x_i^{(1)} \quad (i=0,1), \quad (3)$$

$$y_i^{(1)} = x_i^{(1)} \quad (i=0,1). \quad (4)$$

The nodes in the first layer just transport the signals. The weights of layer $w_i^{(1)} = 1$ ($i$=0,1).

2) The second layer: input term nodes layer.

There are 13+13=26 nodes. Every input node represents one of the 13 fuzzy membership functions. In order to keep symmetry of this controller, the membership functions of 12 inputs $x_i$ ($i$=0,1,2,$\cdots$,5,7,$\cdots$,12) are S functions. The middle one is Gauss function.

$$\text{net}_{ij}^{(2)} = x_i^{(2)} - m_{ij}^{(2)}, \quad (5)$$

$$y_{ij}^{(2)} = 1/[1+\exp(a \times \text{net}_{ij}^{(2)})]$$
$$(i=0,1; j=0,2,\cdots,5)\ (a>0), \quad (6)$$

$$y_{ij}^{(2)} = 1/[1+\exp(-a \times \text{net}_{ij}^{(2)})]$$
$$(i=0,1; j=7,\cdots,12)\ (a>0), \quad (7)$$

$$y_{ij}^{(2)} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{x_i^2}{\sigma^2}) \quad (i=6), \quad (8)$$

where $m_{ij}$ are thresholds of the nodes. They are also the centers of fuzzy memberships. The outputs of this layer are sent to the next layer directly, $w_i^{(3)} = 1$ ($i$=0,$\cdots$,168).

3) The third layer: rule nodes layer.

There are $13 \times 13 = 169$ nodes in this layer. That is the total number of the rules. Every node accomplishes one fuzzy ratiocination operation. Use the AND algorithm to deal with the output of every rule.

$$\text{net}_k^{(3)} = \min[x_i^{(3)}, x_j^{(3)}, 1] \quad (i=0,1,2,3,\cdots,12), \quad (9)$$

$$y_k^{(3)} = \text{net}_k^{(3)} \quad (k=0,1,2,3,\cdots,168). \quad (10)$$

4) The forth layer: output linguistic nodes (Defuzzification) layer.

There is just one output in this layer. The output is the controller force (moment) along the main axis of coordinates, which is normalized.

$$\text{net}^{(4)} = x_i.$$

$$y^{(4)} = (\sum_{i=0}^{168} w_i^{(4)} \text{net}_i^{(4)}) / (\sum_{i=0}^{168} x_i^{(4)}). \quad (11)$$

The weight of neural networks and the positions of S functions are adjusted in this paper. The positions of S functions are expressed with $m_{ij}$ ($i$=0,1; $j$=0,1,$\cdots$,5,7,12). They are set in 26 nodes of the second layer.

# 3 The learning algorithm for advanced FNN

Seen from the network structure, the FNN has the form of multilevel perception, so the adjustable parameters can be corrected by backpropagation. Define the objective function as follows:

$$E = \frac{1}{2}(d - y_1^{(4)})^2 = \frac{1}{2}e^2,$$

where $d$ is the teaching signal. The error signal will be sent back.

1) Defuzzification layer.

$$\delta_1^{(4)} = \frac{-\partial E}{\partial \text{net}_1^{(4)}} = d - y_1^{(4)} = e / \sum x_i^{(4)}, \tag{12}$$

$$\Delta w_i = \frac{-\partial E}{\partial w_i} = \frac{-\partial E}{\partial \text{net}_1^{(4)}} \frac{\partial \text{net}_1^{(4)}}{\partial w_i} = \delta_1^{(4)} \cdot y_i^{(3)}, \tag{13}$$

where $i$=0,1,2,⋯,168.

2) Rule Nodes layer.

$$\Delta w_i = \frac{-\partial E}{\partial w_i} = \frac{-\partial E}{\partial \text{net}_1^{(4)}} \frac{\partial \text{net}_1^{(4)}}{\partial w_i} = \delta_1^{(4)} \cdot y_i^{(3)}, \tag{14}$$

where $j$=0,1,⋯,168.

3) Input Term Nodes layer.

$$\frac{\partial E}{\partial m^{(i)}} = \frac{-\partial E}{\partial y_i^{(2)}} \frac{\partial y_i^{(2)}}{\partial m^{(i)}} = \left( \sum_k \frac{\partial E}{\partial \text{net}_k^{(i)}} \frac{\partial \text{net}_k^{(i)}}{\partial y_i^{(2)}} \frac{\partial y_j^{(2)}}{\partial m^{(i)}} \right) = \left( \sum_k \delta_k^{(3)} \cdot S_{ij}^{(2)} (-y_i^{(2)})(1 - y_i^{(2)})(-a) \right), \tag{15}$$

where ($i$=0,1,⋯,5; $k$=0,1,⋯,5,7,⋯,12; $j$=0,1,⋯,168).

$$\frac{\partial E}{\partial m^{(i)}} = \frac{-\partial E}{\partial y_i^{(2)}} \frac{\partial y_i^{(2)}}{\partial m^{(i)}} = \left( \sum_k \frac{\partial E}{\partial \text{net}_k^{(i)}} \frac{\partial \text{net}_k^{(i)}}{\partial y_i^{(2)}} \frac{\partial y_j^{(2)}}{\partial m^{(i)}} \right) = \left( \sum_k \delta_k^{(3)} \cdot S_{ij}^{(2)} (-y_i^{(2)})(1 - y_i^{(2)}) a \right), \tag{16}$$

where ($i$=7,⋯,12; $k$=0,1,⋯,5,7,⋯,12; $j$=0,1,⋯,168).

When $x_i^{(2)} = \mu_i^j$ is the minimum input value of the $k^{th}$ rule node,

$$S_{ij} = \frac{\partial y_k^{(3)}}{\partial x_i^{(2)}} = \frac{\partial y_k^{(3)}}{\partial \mu_i^j} = 1, \tag{17}$$

or else, $$S_{ij} = \frac{\partial y_k^{(3)}}{\partial x_i^{(2)}} = \frac{\partial y_k^{(3)}}{\partial \mu_i^j} = 0. \tag{18}$$

Then the algorithm can be given for adjusting the parameters.

$$w_i(k+1) = w_i(k) - \eta \frac{\partial E}{\partial w_i}, \tag{19}$$

$$m_i^{(2)} = m_i^{(2)} - \eta \frac{\partial E}{\partial m_i^{(2)}} \quad (i=1,2,\cdots,12), \tag{20}$$

where $a_i$ is constant, which can be chosen according to the range of input variables. In this paper, $a_i$=10.2, $\eta$ is the learning rate of neural network, which can be decided according to experience. It can be either constant or variable[6-12].

# 4 Contrast between advanced FNN and traditional FNN with Gauss membership function

## 4.1 The difference between two FNNs in the membership functions of the input variables

The membership function of the traditional FNN with Gauss membership function:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-m)^2}{\sigma^2}\right).$$

The membership function of advanced FNN:

$$f(x) = \frac{1}{1 + \exp(-a_i(x-m))},$$

where $a_i$ can be changed according to the range of controller parameters.

## 4.2 The differences in property between two membership functions[13]

From the expressions of two functions, it can be known that the Gauss function has the property of radial symmetry. So it has better ability of approximating. And it has higher classification efficiency than S function, especially in higher dimensional space. The space criterion of pattern classification of S function is an open hyperplane, while the space criterion of pattern classification of Gauss function is an open hypersphere, and the pattern recognition accuracy is higher than S function.

But Gauss function is more complicated than S function. The complexity of operation process and training algorithm can be ignored in theoretical research. But in the controller of AUV, the number of samples would be very large. The complication of Gauss function will reduce the incorporated property of controller. However, the expression of S function is simple.

## 4.3 Contrast between trainings of the two networks

The program code of FNN with Gauss membership function is long, and the speed of training is slow. While the program code of advanced FNN with S membership function is short, and only middle part of S function are used in the program, so the speed of training is quick. From the results given in the following, it can be known that when these two kinds of networks are trained with the same samples, the training times of advanced FNN with S membership function is smaller.

## 5 Simulation experiments

In order to validate this algorithm, the advanced FNN with S membership function, the FNN with Gauss membership function, and the simulation program under VC++6.0 environment were established. The object of these two controllers is some min-AUV, which has one propeller, two wings and two rudders.

The samples for training the FNN are shown in Table 1. They are got from maritime trial.

**Table 1 The samples for training**

| Sample | Longitudinal/m | Yaw/(°) | Number |
|---|---|---|---|
| 1 | 10 | 10 | 154 |
| 2 | 10 | 20 | 343 |
| 3 | 50 | 20 | 562 |
| 4 | 80 | 30 | 651 |
| 5 | 80 | 45 | 723 |

In order to accelerate the training of the networks, the distributions of the error and error change are set in the same way.

In this paper, the two FNNs have the same network structure. The only difference between them is the membership function. Before connected to the plan system, these two FNNs have been trained. If taking the training function away from the program, they are fuzzy control.

After training the FNN with S membership function, the distribution curves of its membership functions can be got as shown in Fig.2.

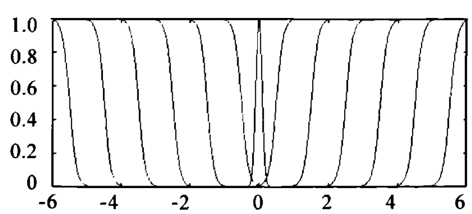


Fig.2 Distribution of the membership function

In the movement simulation, the objective position of longitudinal position was 5 m. The objective position of yaw was 30°. The initial position was 0. There's no current. The results are shown in Figs.3 ~ 4.

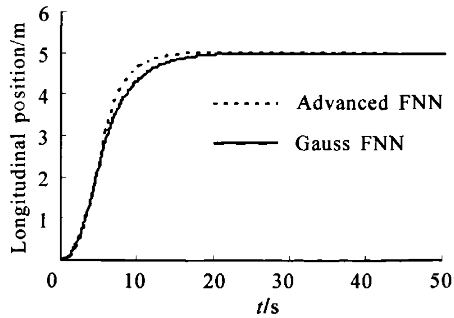


Fig.3 Longitudinal motion control curves of two FNNs

Because Gauss function has the property of radial symmetry and approximation of function, the motion curve of the FNN with Gauss membership function is very smooth and the variance of motion is very slow. With the change of position error, the velocity modulability of Gauss membership function-based controller is better.

Because the training speed of S functions is high, the response of this controller is fast. It makes the AUV get the objective position earlier. Because of the distribution of its membership function, when the velocity reaches a relative stable value, it is not easy for this controller to change its velocity according to the change of position error. That means its velocity stability is well. Such velocity stability makes it unavoidable to produce overshoot. But from the results of experiment, it can be known that it can keep AUV stable at objective point at last. When the objective position is 5 m, the overshoot is 1.014 2%. In practice, 5% overshoot is permitted.

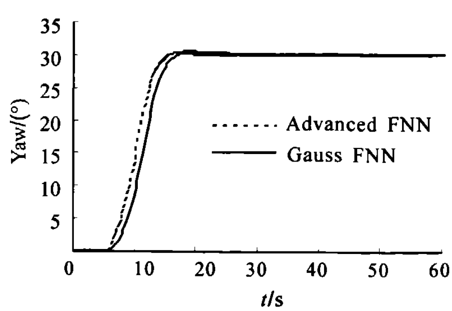


Fig.4 The yaw motion control curves of two FNNs

In the beginning, the line is horizontal because the operation of rudders is limited by the velocity of AUV.

When the velocity of AUV is lower, the lift forces of rudders are very small. Then the effect of rudders is not obvious, and they will add the resistance of AUV, so the operation of rudders was confined. Only when the velocity of AUV gets a certain value can the signals be sent to set certain angles of rudders for changing the yaw of AUV.

As there is a Gauss function in the middle, at first the advanced FNN with S membership function has similar acceleration with the FNN with Gauss membership function. Because of the property of S function, it will catch up with the plan system faster than the FNN with Gauss membership function, and it will get near to the objective point quickly. Just because of the reasons analyzed above, the velocity stability makes it unavoidable to produce overshoot. From the experiment, conclusions can be drawn that when the objective yaw is 30°, the overshoot is 1.933%. Its stable error is 0. It can satisfy the real application.

# 6 Conclusions

Because of the task property of this autonomous underwater vehicle, it doesn't need to have the ability of positioning accurately. It only needs to respond quickly when coming across obstacles, so that it can voyage in the ocean safely. The advanced FNN with S membership function can satisfy this requirement. Furthermore, because its algorithm is simple, its program code is shorter and the intermediate data are fewer correspondingly, it requires less from the hardware of embedded computer (such as internal storage capacity). From the results of simulation, it can be known that the advanced FNN with S membership function can achieve the favorable result in the AUV controller.

# References


[1] LI Ye, LIU Jiancheng, SHEN Mingxue. Dynamics model of underwater robot motion control in 6 degrees of freedom[J]. Journal of Harbin Institute of Technology, 2005, 12(4): 456-459.

[2] XING Jie, XIAO Deyun. S The application of membership function-based falcon[J].Journal of Harbin Institute of Technology, 2006, 27 (suppl.): 22-26(in Chinese).

[3] YI Kaigu, HOU Aibin. Intelligent control[M]. Beijing: Press of Beijing University of Technology, 1999: 177-234(in Chinese).

[4] LIU Jinkun. Intelligent control[M]. Beijing: Electronic Industry Press, 2005:18-150(in Chinese).

[5] WANG J S, LEE C S G, YUH J. Self-adaptive neuro-fuzzy control with fuzzy basis function network for autonomous[C]// Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Victoria, 1999:130-135.

[6] PENG Liang, LU Yingchun, WAN lei. Neural networks control of autonomous underwater vehicle[J]. Ocean Engineering, 1995, 13(2): 38-46(in Chinese).

[7] WANG L X, MENDEL J M. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning[J]. IEEE Trans Neural Networks, 1992, 3(5): 807-814.

[8] SERRANI A, CONTE G. Robust nonlinear motion control for AUVs[J]. IEEE Robotics & Automation Magazine, 1999 ,6(2): 33-38.

[9] ISHII K, FUJII T, URA T. Neural network system for on-line controller adaptation and its application to underwater robot[C]// Proc of 1998 IEEE Int Conf on Robotics and Automation. Leuven, 1998(5): 756-761.

[10] WANG J S, LEE C S G, YUH J. An on-line self-organizing neuro-fuzzy controller for autonomous underwater vehicles[C]// Proc of 1999 IEEE Int Conf on Robotics and Automation. Detroit: Omnipress, 1999: 2416-2421.

[11] WANG J S, LEE C S G. Structure and learning in self-adaptive neural fuzzy inference systems[J]. International Journal of Fuzzy Systems, 2000, 2(1):12-22.

[12] LIN C T, LEE C S G. Neural- network-based fuzzy logic control and decision system[J]. IEEE Trans Comput, 1991, 40(12): 1320-1336.

[13] WANG Shoujue, WANG Bonan. Multidimensional space geometric analysis and its theory of neural network[J]. Journal of Electronic, 2002, 30(5): 1-4(in Chinese).


**XU Yu-ru** was born in 1943. He is the Chinese Academy of Engineering (CAE) academician, a professor and PhD supervisor at Harbin Engineering University. His current research interests include intelligent control and motion simulation of underwater vehicle.

**GUO Bing-jie** was born in 1982. She is a PhD student at Harbin Engineering University. Her current research interests include intelligent control and motion simulation of underwater vehicle.

# Advanced FNN control of mini underwater vehicles

作者： XU Yu-ru， GUO Bing-jie， LI Yue-ming
作者单位： College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China
刊名： 哈尔滨工程大学学报(英文版)
英文刊名： JOURNAL OF MARINE SCIENCE AND APPLICATION
年，卷(期)： 2008，7(3)


## 参考文献(13条)

1. LI Ye.LIU Jiancheng.SHEN Mingxue Dynamics model ofunderwater robot motion control in 6 degrees of freedom[期刊论文]-Journal of Harbin Institute of Technology 2005(04)

2. XING Jie.XIAO Deyun.S The application of membershipfunction-based falcon 2006(zk)

3. YI Kaigu.HOU Aibin Intelligent control 1999

4. LIU Jinkun Intelligent control 2005

5. WANG J S.LEE C S G.YUH J Self-adaptive neuro-fuzzycontrol with fuzzy basis function network forautonomous 1999

6. PENG Liang.LU Yingchun.WAN lei Neural networkscontrol of autonomous underwater vehicle 1995(02)

7. WANG L X.MENDEL J M Fuzzy basis functions,universal approximation,and orthogonal least-squareslearning 1992(05)

8. SERRANI A.CONTE G Robust nonlinear motion controlfor AUVs 1999(02)

9. ISHII K.FUJII T.URA T Neural network system for on-linecontroller adaptation and its application to underwaterrobot 1998

10. WANG J S.LEE C S G.YUH J An on-linself-organizing neuro-fuzzy controller for autonomousunderwater vehicles 1999

11. WANG J S.LEE C S G Structure and learning inself-adaptive neural fuzzy inference systems 2000(01)

12. LIN C T.LEE C S G Neural-network-based fuzzy logiccontrol and decision system 1991(12)

13. WANG Shoujue.WANG Bonan Multidimensional spacegeometric analysis and its theory of neural network 2002(05)

## 相似文献(0条)